Unstructured multigridding by volume agglomeration : current status

Marie-Hélène Lallemand, Hervé Steve, Alain Dervieux

UNITÉ DE RECHERCHE
INRIA-SOPHIA ANTIPOLIS

Institut National
de Recherche
en Informatique
et en Automatique

Domaine de Voluceau
Rocquencourt
B.P.105
78153 Le Chesnay Cedex
France
Tél.:(1) 39 63 55 11

Rapports de Recherche

N° 1224

Programme 7
Calcul Scientifique,
Logiciels Numériques et Ingénierie Assistée

UNSTRUCTURED MULTIGRIDDING BY VOLUME AGGLOMERATION: CURRENT STATUS

Marie-Hélène LALLEMAND
Hervé STEVE
Alain DERVIEUX

Mai 1990

RR-1224

UNSTRUCTURED MULTIGRIDDING BY VOLUME AGGLOMERATION: CURRENT STATUS

Marie-Helene LALLEMAND, Herve STEVE, Alain DERVIEUX
INRIA, 2004 Route des Lucioles, 06560 VALBONNE , FRANCE

MULTIGRILLES EN MAILLAGE NON STRUCTURE PAR VOLUMES FINIS AGGLOMERES : UN BILAN

Marie-Helene LALLEMAND, Herve STEVE, Alain DERVIEUX
INRIA, 2004 Route des Lucioles, 06560 VALBONNE , FRANCE

ABSTRACT :

We describe a multigrid method for solving the Euler equations applying to non-structured meshes in 2-D (triangles) and 3-D (tetrahedra). The main idea is to coarsen the given mesh by using topological neighboring relations. It is applied to upwind solvers relying on the MUSCL methodology. Two multi-grid schemes are presented: an explicit Runge-Kutta FAS one and an implicit Correction Scheme. Transonic external flows computations are described for illustration.

RESUME :

On décrit une méthode multigrille pour résoudre les équations d'Euler en maillage non structuré 2-D ou 3-D. L'idée principale consiste à générer des maillages grossiers en agglomérant des volumes finis voisins. Cette approche est appliquée à des méthodes décentrées de type MUSCL. Deux versions sont présentées : une version explicite FAS avec iteration de type Runge Kutta, une version implicite de type "Correction Scheme". Quelques calculs d'écoulements transsoniques sont présentés à titre d'exemple.

INTRODUCTION :

The interest for unstructured meshes in CFD appears more and more stronger because better mesh generators and adaptators are developed.

Concerning the flow solvers, we think that, by comparison with "structured solvers", the main question is related to **efficiency**. Many efficient methods developed for the structured case are not easily extended to the unstructured one and necessitate many researchs in this direction. One subject is the extension of Multi-Grid (MG) schemes; this paper is essentially devoted to it: the "unstructured aspects" of multigridding on finite-element meshes will be considered for the solution of compressible inviscid flow calculations; we shall focus on firstly the method used to generate the different grids and secondly the solution algorithms to be combined with the MG approach.

Let us discuss the existing methods for generating the different levels; two families of methods can be considered :

Topological methods are often based on mesh enrichment; starting from an unstructured coarse grid, finer grids are generated by element division either over the whole computational domain or locally only, after a posteriori error estimates. We refer to [1], [2], [3], [4], [5], [8] for studies using these two points of view. These topological methods cannot handle the case of an a priori given fine grid ; then the quality of the fine grid is too much dependent on the initial coarse grid.

Another topological approach consists in generating separately non embedded grids ([6], [7], [8]).

In algebraic methods, starting from a linear system derived from an arbitrary unstructured fine mesh formulation, coarser levels are generated by gathering related equations (lines of the matrix) ; we refer to [9]. Algebraic Multigrid Methods are usually applied to linear systems and necessitate the construction and generally also the storage of the matrix.

In this paper, we propose a new topological approach with the following features :
- the grids are embedded ;
- the coarse meshes are not classical FEM triangulations but generalized finite volume partitions ;
- the spatial approximation is derived on each level.

Once the different levels have been obtained, we must choose some **basic schemes** to iterate on each grid towards the solution. In this paper, we focus on explicit or explicit-like nodewise iteration; our motivations are of two types: firstly a complex iterative algorithm may need much storage, secondly we wish to use vector (and eventually parallel) computers. The nodewise iterations can be the usual time-stepping in the context of explicit schemes or the Jacobi iteration in implicit schemes. In both cases, multi-step iterations are considered, in order to optimize the convergence properties of the algorithm.

The plan is the following:
1. Spatial upwind approximation
2. Explicit and implicit one-grid solvers
3. Toward MG: generation of the coarse levels
4. Generalized finite-volume upwind scheme
5. Explicit multi-grid scheme
6. Implicit multi-grid scheme
7. Conclusion

1. SPATIAL UPWIND APPROXIMATION

The object of this section is to recall some methods for the spatial discretization over a finite element simplicial mesh (triangles, tetrahedra) of the Euler equations. These methods have been introduced in [10] ; see also [11]. Ideas both from finite volume and finite element methods are combined ; since the basic ideas were introduced by B. Van Leer in the construction of MUSCL, we shall refer to the schemes presented here as MUSCL-FEM schemes.

1.1. GOVERNING EQUATIONS

The conservative law form of the Euler equations in the spatially two-dimensional case is given by :

$$W_t + \frac{\partial}{\partial x} F_1(W) + \frac{\partial}{\partial y} F_2(W) = 0 \tag{1}$$

With

$$W = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ E \end{pmatrix} = (W^k)_{k=1,4} \tag{2}$$

and :

$$F_1(W) = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ u(E+p) \end{pmatrix}, \; F_2(W) = \begin{pmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ v(E+p) \end{pmatrix} \tag{3}$$

where ρ is the density, $\vec{U} = (u, v)$ is the velocity vector, E is the total energy per unit volume and p is the pressure. We assume that the fluid satisfies the perfect gas law :

$$p = (\gamma - 1)\,(E - \frac{1}{2}\,\rho||\vec{U}||^2)$$

where γ , ratio of specific heat, to be equal to 1.4 in the air.

Using a symbolic notation for the vectors $(F_i)_{i=1,2}$ in (3),

$$\vec{F}(W) = \begin{pmatrix} F_1(W) \\ F_2(W) \end{pmatrix}$$

then the conservative system (1) becomes :

$$\frac{\partial W}{\partial t} + \vec{\nabla} \cdot \vec{F}(W) = 0.$$

sometimes referred to as the "divergence" form.

The boundary conditions are the following: wall condition is a slip condition; farfield condition is that the flow is uniformly equal to a given one.

1.2. SPATIAL APPROXIMATIONS IN 2-D

1.2.1. Triangulation and volumes

We assume that the computational domain Ω is a polygonal bounded domain of $\mathbb{R}^2$. Let Θ_h a standard [12] finite element triangulation of Ω and h the maximal length of the sides in Θ_h .

We introduce the **dual finite-volume mesh** :

For every vertex S_i $(i = 1, ..., n_h)$ of Θ_h , we define a cell C_i as follows

- every triangle having S_i as a vertex is subdivided in 6 subtriangles by means of the medians.
- the cell C_i is the union of the resulting subtriangles having S_i as a vertex. The boundary of C_i is denoted by ∂C_i (Fig. 1).

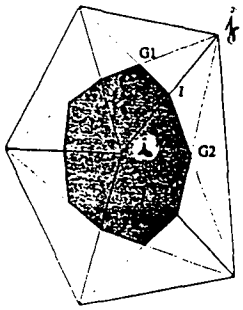


FIGURE 1 :
Dual Cell around a vertex

We also introduce the following notations

- $K(i)$ is the set of indices of neighboring nodes of S_i.
- $\vec{\nu}_i = (\nu_{ix}, \nu_{iy})$ is the unit vector of the outward normal to ∂C_i .

1.2.2. First-order accurate approximations

The two-dimensional extension of the class of three-points upwind first-order accurate 1-D schemes is done as follows : using Green's formula, the Finite-Volume formulation of (1) can be written

$$(4) \qquad area(C_i)\ \frac{W_{h_i}^{n+1} - W_{h_i}^{n}}{\Delta t} + \int_{\partial C_i} \vec{F}(W_h^n)\ .\ \vec{\nu}_i\ d\sigma\ =\ 0\ .$$

To compute the integral arising in (4) , we split the surface ∂C_i into panels ∂C_{ij} , $(j \epsilon K(i))$, separating node S_i and node S_j (see Fig.2).

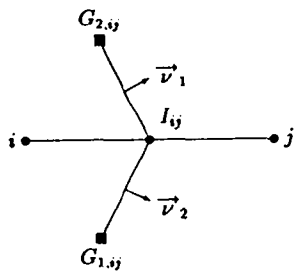


FIGURE 2 :
Portion of a cell boundary, separating node S_i from node S_j .

Then the discrete problem consists of finding, at every new time level t^{n+1}, the vector W^{n+1} defined by the discrete system :

$$area(C_i)\ \frac{W_i^{n+1} - W_i^{n}}{\Delta t} + \sum_{j\ \epsilon\ K(i)} \int_{\partial C_{ij}} \vec{F}(W^n)\ .\ \vec{\nu}_i\ d\sigma$$

$$+ \int_{\partial C_i \cap \Gamma_B} \vec{F}(\bar{W}^n)\ .\ \vec{\nu}\ d\sigma + \int_{\partial C_i \cap \Gamma_\infty} \vec{F}(\bar{W}^n)\ .\ \vec{\nu}\ d\sigma\ =\ 0$$

Γ_B holds for the profile boundary, and Γ_∞ for the farfield one ; the notation $\bar{W}^n$ will be explained in the sequel ; we now specify the computation of the integral over ∂S_{ij} ; actually, the evaluation of this

term corresponds to the one-dimensional calculation of the flux along the direction S_iS_j , that will be computed from the following metric coefficients:

$$\vec{\eta}^{ij} = \int_{\partial C_{ij}} \vec{\nu}_i \, d\sigma$$

For upwinding, we introduce a numerical flux function ; the approximate system becomes :

$$\begin{aligned} & area(C_i) \, \frac{W_i^{n+1} - W_i^n}{\Delta t} + \sum_{j \in K(i)} \Phi(W_i^n, W_j^n, \vec{\eta}^{ij}) \\ & + \int_{\partial C_i \cap \Gamma_B} \vec{F}(\bar{W}_h^n) \cdot \vec{\nu} \, d\sigma + \int_{\partial C_i \cap \Gamma_\infty} \vec{F}(\bar{W}_h^n) \cdot \vec{\nu} \, d\sigma = 0 \end{aligned} \tag{5}$$

where $\Phi(U, V, \vec{\eta})$ denotes a numerical flux splitting : we choose Osher's splitting [13] for its good stability properties when applied to high Mach number flow simulation [14]. In addition, it results in a differentiable flux function, which can be crucial in correctly capturing stagnation zones and sonic points, where the Jacobian matrices are singular. This flux function is given in short by :

$$\begin{aligned} F_{ij} &= \vec{\eta}_x^{ij} \, F_1 + \vec{\eta}_y^{ij} \, F_2 \\ \Phi^{OSHER}(U, V, \vec{\eta}^{ij}) &= \frac{F_{ij}(U) + F_{ij}(V)}{2} - \frac{1}{2} \int_U^V |F_{ij}{}'(W)| dW \end{aligned}$$

The integration from U to V is performed successively along three paths defined by the Riemann invariants ; these paths are uniquely defined if we except symmetry ; the signs of eigenvalues are taken into account for exact integration. We have also used the very robust and simple Flux Vector Splitting of van Leer [15].

Integrals over Γ_B and Γ_∞ in (5) involve the physical boundary conditions, that are taken into account through the vector $\bar{W}^n$: this vector is computed from quantities depending on the node values W^n and quantities derived from the physical boundary conditions

For a wall boundary integral, the vector $\bar{W}^n$ is defined as satisfying the slip condition and thus the integral over Γ_B is written :

$$\int_{\partial C_i \cap \Gamma_B} \vec{F}(\bar{W}^n) \cdot \vec{\nu} \, d\sigma = \int_{\partial C_i \cap \Gamma_B} \begin{pmatrix} 0 \\ \bar{p} \, \nu_x \\ \bar{p} \, \nu_y \\ 0 \end{pmatrix} d\sigma \, ,$$

where $\bar{p}$ is equal to the node pressure $p(W^n)$. With this procedure, the slip condition is applied in a weak variational way.

For inflow and outflow boundary integrals, we have to select a precise set of exterior datas, depending on the flow regime and the velocity direction ; for this purpose, a plus-minus splitting is applied between exterior datas and interior (=node) values.

1.2.3. Second-order accurate approximations

We recall in short in this section the derivation of the MUSCL-FEM schemes [10], [11]. With constant-by-cell dependent variables, the above numerical split-flux integration will result in schemes which are only first-order accurate.

The main ingredients in constructing a (non-oscillatory) second-order accurate approximation are the following ones :

(i) the above first-order quasi-monotone scheme with upwind flux function ;

(ii) a second-order variant using linear interpolations ;

(iii) a slope-limiting procedure to reduce the oscillations in the solution.

This approach intends to extend Van Leer's MUSCL method [16] to non-structured meshes. In this paper, several types of second-order non-oscillatory extensions have been used; we just describe a typical one, that is referred as an Hermitian slope limited formulation. This kind of slope limiter directly acts in the computation of approximate node gradients of flow variables ; the employed flow variables are the density, the velocity components and the pressure :

$$\hat{W} = (\rho, u, v, p)$$

The approximate x-derivative, for example, is computed as the value of least modulus among the Galerkin x-derivatives in the neighboring triangles if they are all of the same sign, and as zero otherwise :

$$\frac{\partial \hat{W}^{lim}}{\partial x}(i) = \min_{T \text{ neighbor of } i} \text{mod} \left(\frac{\partial \hat{W}}{\partial x}|_T \right). \tag{9}$$

This is applied in the same way for boundary nodes. These quantities are then used for the interpolation of the flow variables :

$$\hat{W}_{ij} = \hat{W}_i + \frac{1}{2} \begin{pmatrix} \frac{\partial \hat{W}^{lim}}{\partial x}(i) \\ \frac{\partial \hat{W}^{lim}}{\partial y}(i) \end{pmatrix} . \vec{ij}$$

$$\hat{W}_{ji} = \hat{W}_j + \frac{1}{2} \begin{pmatrix} \frac{\partial \hat{W}^{lim}}{\partial x}(j) \\ \frac{\partial \hat{W}^{lim}}{\partial y}(j) \end{pmatrix} . \vec{ji}$$

From this conservative variables W_{ij} and W_{ji} are derived and used in the split-flux ; this results in the following discrete equation :

$$\frac{area(C_i)}{\Delta t}(W_i^{n+1} - W_i^n) + \sum_{j \in K(i)} H_{ij}^{(2)} + \int_{\partial C_i \cap \Gamma_B} \vec{F}(\bar{W}^n) . \vec{\nu} d\sigma + \int_{\partial C_i \cap \Gamma_\infty} \vec{F}(\bar{W}^n) . \vec{\nu} d\sigma = 0, \tag{10}$$

with

$$H_{ij}^{(2)} = \Phi(W_{ij}, W_{ij}, \vec{\eta}^{ij})$$

For a different slope limiter, we refer to [11].

1.3. SPATIAL APPROXIMATION IN 3-D

Every concept of the 2-D construction is easily extended to 3-D ; let us make more precise some geometrical points :

The computational domain Ω is assumed to be a polyhedral bounded subdomain of $I\!R^3$, and Θ_h is a standard finite element tetrahedrization, with h as maximal length of the edges in Θ_h.

A **dual finite volume partition** is derived from the construction of median planes, that is, for every vertex S_i of Θ_h, we define a cell or finite volume C_i "around S_i" as follows

- every tetrahedron having S_i as a vertex is subdivided in 24 subtetrahedra by means of planes containing an edge and the middle of the opposite edge ; then the cell C_i is the union of subtetrahedra having S_i as a vertex.

In particular, the boundary ∂C_i of C_i is the union of the $\partial C_{ij} = \partial C_i \cap \partial C_j$ that can be defined as the union of triangles such that (Fig. 3-4)

- one vertex is the middle of the edge S_iS_j
- one vertex is the centroid of the tetrahedron T having S_iS_j as an edge
- one vertex is the centroid of a (triangular) face of T having S_iS_j as a side.

Then the other feactures of the 2-D deviation clearly extend to 3-D, for a different slope limiter, we refer to (14).

FIGURE 3 :
Finite volume partition in 3-D:
intersection of C_i with an element T

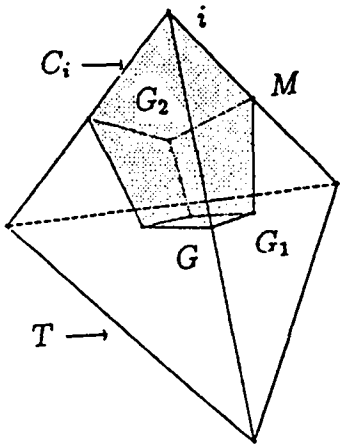
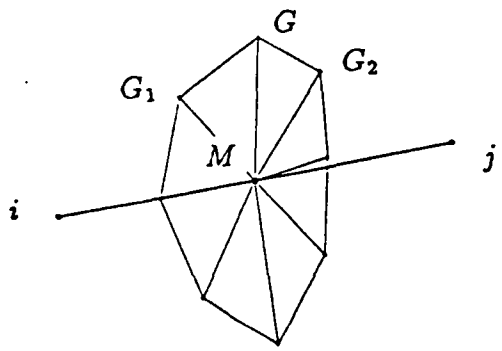


FIGURE 4 :
Finite volume partition in 3-D:
intersection of C_i and C_j



i : node

T : tetrahedron

C_i : cell around i

M : middle of edge

G : face centroid

G_1, G_2 : face centroids

2. EXPLICIT AND IMPLICIT ONE-GRID SOLVERS

We use pseudo-unsteady solution algorithms, that can be explicit or implicit in time.

2.1. Explicit schemes

Following A. JAMESON [17], a Runge-Kutta scheme is applied with either one or four stages.

Let define $E_k(W)$ the flux terms where k is the order of the accuracy of the interpolation: $k = 1$ (first order accurate) or $k = 2$ (MUSCL formulation). We write the L-stage Runge-Kutta algorithm as follows:

$$\begin{aligned} & W^{(0)} = W^n; \\ & \textit{for } l = 1 \textit{ to } L \textit{ do}: W^{(l)} = W^{(0)} - \Delta t\, \alpha_l\, E_k(W^{(l-1)}); \\ & W^{n+1} = W^{(L)}. \end{aligned} \tag{11}$$

When the four-stage algorithm is applied, the following coefficients are employed (see [18], [19]) :

$$\alpha_1 = .11 \quad ; \alpha_2 = .2766 \quad ; \alpha_3 = .5 \quad ; \alpha_4 = 1. \tag{12}$$

Explicit methods are very robust but not enough efficient to reach the steady state; implicit methods can remedy this, by applying large time steps.

2.2. Low-storage implicit schemes

Implicit unfactored relaxation schemes are more and more attractive ; for unstructured mesh calculation, the use of such schemes seems to be one among the most efficient algorithms [11].

We describe now an implicit code that presents already a rather satisfactory behavior in a large range of Mach numbers (from Mach = 10^{-3}, see [20], to Mach = 25, see [14]),

We calculate the solution at each time-step in two phases.

Firstly, an explicit phase is performed :

$$\widehat{\delta W} = -\Delta t E_k(W^n) \tag{13}$$

where Δt is the time-step and E_k the numerical flux.

Secondly, an implicit phase is applied. The implicit terms use a matrix A^n assembled from first-order accurate fluxes, and which comes from a linearization at W^n of the Steger-Warming decomposition [11].

$$X = (A^n)^{-1}\, \widehat{\delta W} \; ; \; W^{n+1} = W^n + X. \tag{14}$$

Therefore at each time-step we have to solve a linear problem; the way to solve it efficiently without using too much memory storage is introduced in [21].

Jacobi low storage iteration :

Algorithm : When heavy geometries (many nodes) have to be taken into account, the matrix of the linear operator cannot be stored; an attractive trade-off consists in storing diagonal terms and recomputing the extra-diagonal ones. Let D^n denote the block-diagonal part of A^n; the above linear system is partly solved by a few block-collective Jacobi iterations ($kmax = 4$ here) :

$$X^0 = (D^n)^{-1}\widehat{\delta W}$$

$$\text{for } k = 1 \text{ to } kmax,$$

we do

$$X^{k+1} = (D^n)^{-1}(\widehat{\delta W} - (A^n - D^n)X^k) \tag{15}$$

and

$$W^{n+1} = W^n + X^{kmax}.$$

The inverse $(D^n)^{-1}$ of the block diagonal part of the matrix is stored, while the other term $A^n - D^n$ is not stored, but calculated at each Jacobi iteration. There are two main reasons for the choice of the Jacobi method : first, the algorithm is not recurrent (unlike the Gauss-Seidel method), therefore independency is preserved for vectorization. Secondly, the low storage version is derived more easily. When a scalar computer is used, this method is less efficient than the implicit full-storage method, but still 5 to 12 times more than an explicit scheme (3-D tests). For vector computers, the loss of efficiency w.r.t. the Jacobi full-storage method is lower since extra calculations do not involve many scatter and gather and are therefore fastly vectorizable.

Memory storage : In the 3-D case, the total matrix storage requires $350 \times NS$ real numbers (NS is the total number of nodes) with only $25 \times NS$ for the diagonal terms. (We need $100 \times NS$ words for the rest of storage).

The efficiency of the scheme for steady-state calculations is related to its convergence properties: the above time-iteration is near of Newton's method when only first-order approximations are applied in both explicit and implicit phase; for the case of the first-order preconditioning of a second-order approximation, we refer to the work of Desideri [22] which shows that a convergence ratio of .5, independently of the mesh, can be ideally reached.

3. TOWARDS MG: GENERATION OF THE DIFFERENT LEVELS

The objective is to generate coarse levels automatically from an arbitrary unstructured triangulation.

To achieve such degree of reliability, we explore the possibility of grouping together nodes associated with contiguous control volumes starting from the **dual** finite-volume mesh. Repeating the operation allows us to get coarser and coarser levels until sufficiently many levels are obtained.

An algorithm for grouping cells together should satisfy the following criteria :

(1) The size of the mesh should decrease while the maximum allowable time step (for explicit iterations) should increase.

(2) The solution should be sufficiently accurately represented on coarser grids to obtain a good initialization (Full-MG) and good preconditioners ,

(3) The sequence of nested grids should allow the damping of a dense enough collection of frequency modes,

(4) The procedure should not be costly.

One approach could consist of using some auxiliary regular coarser mesh which divides the domain in regions, in order to gather the cells whose centers belong to the same region. Such an approach may not sufficiently account for the node density of the initial mesh. Some more sophisticated methods could be considered : we could derive them from the works motivated by multi-tasking in super-computers ; the problem is to divide the domain in regions which are (1) of comparable size (number of nodes) and (2) with as few connections between each other as possible (therefore with as much connection in each region). The sophistication can be increased up to the examination of the discrete system as in Algebraic MG methods.

The coarsening algorithm that we present here is only based on direct-neighboring relations (two cells are direct-neighbors if they contain vertices that are neighbors) and reads as follows :

Coarsening algorithm:

Consider successively every cell.
(1) if the cell C is already included in a group then consider next cell; else: create a new group containing C, and put into this group each cell neighboring C and not already included in a group.
(2) if the new group contains only the cell C, then destroy the group and put cell C in an existent group containing a neighbor of C.
(3) next cell.

This algorithm has been written [19] in order to allow a efficient and transparent automated coarsening up to a two-cell coarsest grid. The cost of the coarsening step was in the experiements always under one purcent of the flow calculation.

4. GENERALIZED FINITE VOLUME UPWIND SCHEMES

One main feature of the algorithm is that it relies on the construction of a Finite-Volume Method applicable to an arbitrary partition of the computational domain.

4.1. Coarse mesh upwind scheme

The upwind finite volume scheme is derived in the simplest manner that one can imagine. We describe it in the context of the usual explicit time-stepping. Given a cell C_i , the mean value W_i of the dependent variable in this cell is advanced from time level n to time level $n+1$ as in (5).

On the finest mesh, that is a standard triangulation, either the first order scheme, or the second order scheme defined in Section 1 is applied.

4.2. Stability and time-step length

The efficiency and the robustness of the algorithm relies on the accurate estimation of the maximal local time-step ; this is particularly essential when **local time-stepping** is employed. Unfortunately, trying to estimate the local stable time-step for a coarse mesh by generalising a simplified Fourier analysis can be very hazardous. Hence, we prefer to evaluate a lower bound based on the L^∞ stability of a multi-dimensional non regular model. We consider a conservative linear scalar model:

$$u_t + div(\vec{V}u) = 0.$$

where $\vec{V}$ is given but not constant, $\vec{V} = \vec{V}(x,y) = (V^x, V^y)$.

It is reasonable to consider that a numerical scheme which approximates this conservation law is stable if it preserves the positiveness of the dependent variable. The conservative scheme is derived :

$$area(C_i)(u_i^{n+1} - u_i^n) = -\Delta t \sum_{j \ neighbor \ of \ i} \alpha_{ij}(\theta_{ij}u_i^n + (1-\theta_{ij})u_j^n)$$

with

$$\alpha_{ij} = \eta_x^{ij}(\frac{V_i^x + V_j^x}{2}) + \eta_y^{ij}(\frac{V_i^y + V_j^y}{2}) \ ; \ \theta_{ij} = \frac{1}{2}(sign(\alpha_{ij}) + 1)$$

LEMMA 1 : *The above scheme preserves the positiveness of its solution if the following inequality holds for every cell C_i :*

$$area(C_i) - \Delta t \max_{\text{j neighbor of i}} \|\vec{V}_j\| \int_{\partial C_i} d\sigma \geq 0.$$

Adaptation to the Euler scheme :

Using the notations of Section 1, the following "reference time-step" will be computed on each cell as follows:

$$\Delta t_i = area(C_i)/(\lambda^i_{\max} \int_{\partial C_i} d\sigma)$$

with

$$\lambda^i_{Max} = Max(\lambda_i, \max_{\text{j neighbor of i}} \lambda_j)$$

and

$$\lambda_i = (u_i^2 + v_i^2)^{\frac{1}{2}} + c_i$$

where u_i, v_i, c_i hold for the values in cell C_i of (resp.) horizontal velocity, vertical velocity, and sound speed. In practice, time-step larger than Δt_i by a factor of 3 in the 2-D case (resp. 6 in the 3-D case) can be used (L^2-)stably and a good strategy for multi-gridding is to set Δt in the range of $2.5\Delta t_i$ to $3.\Delta t_i$ in 2-D, the factor being between 5 and 6 in 3-D.

5. EXPLICIT MULTIGRID SCHEME

5.1. FMG first-order scheme

The basic iteration is the above explicit Runge-Kutta time-stepping. The MG algorithm uses FAS iterations and Full Multigridding as in [17]. Saw-tooth V-cycles are applied. We have sketched it in Fig. 5 for the three-grid case. The transfer operators are defined in the present approach as follows :

- Fine-to-coarse : values are averaged in a conservative manner.
- Coarse-to-fine : the trivial injection is applied.

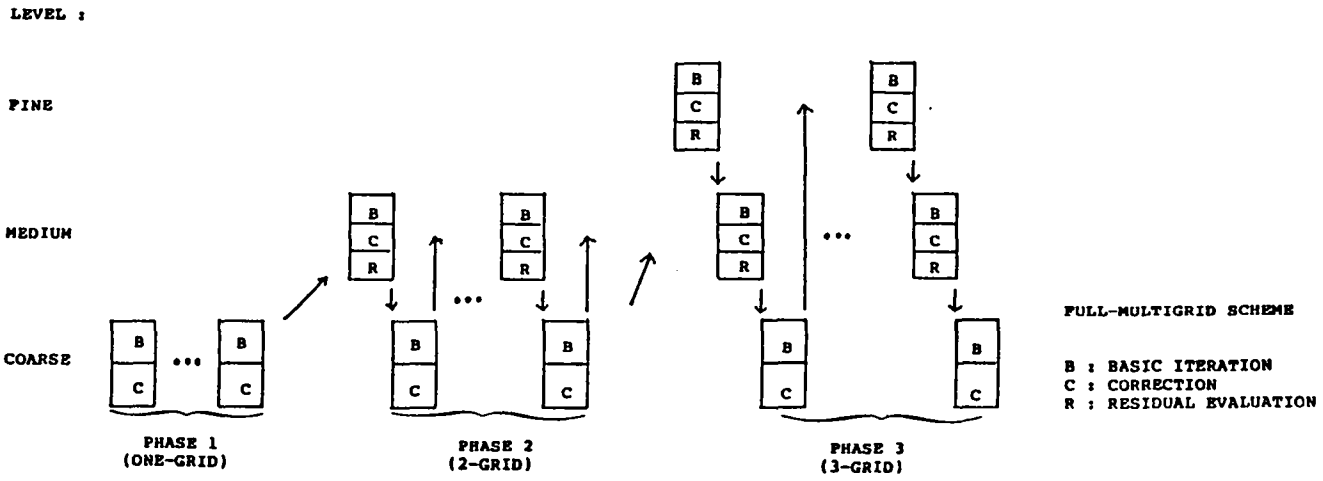


FIGURE 5
Saw-tooth V-cycle

We keep the coefficients defined in (12) since they where proved in [19] to be somewhat optimal for high frequency damping.

5.2. Second-order version

The second-order spatial scheme is introduced into the fine-grid solver only for the last phase of the full-multigrid process. This introduces a minor modification in the algorithm. However, two disadvantages appear in this construction : first the coarse level correction is less consistent with the fine level smoother ; second, in a full-multigrid approach, the last phase starts from a first-order (medium level) solution instead of a second-order one.

Another possibility is to apply a Defect Correction approach [23]; we refer to [24] for such a work with structured meshes and to [25] for an DeC extension of the present MUSCL-FEM MG scheme.

5.3. NUMERICAL ILLUSTRATIONS

5.3.1. Two experiments with nested meshes

The calculations of an internal flow in a channel with a 4.2 % thick bump are presented. It has been observed that the regime defined by a Mach number at infinity equal to .85 is representative of the usual stiffness of transonic flow calculation.

A first mesh is presented in Fig.6 and contains 161 vertices. The three successive levels are also depicted : the dual fine level (161 control volumes), the medium level, the coarse level. The convergence histories are shown for -a one-grid calculation over the successive levels using as initial data the result obtained from the previously employed coarser grid, - a full-multi-grid calculation, that is one-grid scheme with the coarse level, then two-grid scheme with the medium one, and three-grid with the fine triangulation. The one-step Runge-Kutta scheme is used. In the third phase, the residual reduced by three orders of magnitude after 159 cycles. In order to evaluate the behavior of the scheme when the number of nodes is increased, we present the same experiment with a finer triangulation derived from the previous one by dividing equally each triangle into four new ones. The triangulation now contains 585 vertices (Fig. 7) ; we present a first-order four-grid FMG calculation ; in the fourth phase, the residual is reduced by three orders of magnitude after 155 cycles (see Fig. 7).

5.3.2. Application to a strictly non-structured mesh

The efficiency is now evaluated with the calculation of an external flow around a NACA0012 airfoil (Mach =.72, angle of attack = 0 deg.). The triangulation is now really unstructured : it results from the use of a mesh generator based on a front advancing algorithm and contains 800 vertices (Fig.8a). The convergence is again fast ; the second order accurate solution is obtained in about 150 three-grid cycles in the third phase when the one-step Runge-Kutta is applied (Fig. 8f) while the convergence of the first-order version required only 80 cycles (Fig.8e). The loss of symmetry in the coarse grids does not seem severely penalizing. The Mach and pressure contours of the resulting solution are shown in Fig. 8g.

5.3.3. Comparison with a classical approach

We consider the same test case as in 5.3.2, and we want to compare our scheme with (what we call) **a multi-triangulation algorithm** : three triangulations are nested standardly (by element division) and as spatial scheme, the second-order upwind scheme can be applied at each level. The successive nested triangulations contain respectively 121, 442, and 1684 vertices. We present three calculations relying on the fine triangulation, using **four-step Runge-Kutta** three-grid algorithms and starting from uniform flow. A comparison of the convergence histories of the two algorithms is presented in Fig.5e :

- when the multi-triangulation algorithm is applied with second order flux-splitting over the three levels, the solution is obtained in about 40 cycles.
- when the same algorithm is applied, but with first order flux-splitting over the two coarse levels and the second order splitting over the fine level, the solution is obtained also in about 80 cycles but with a more monotone convergence.
- with the presented algorithm, the solution is obtained in about 80 cycles (with the same convergence).

This seems to prove that the difference between these two approaches essentially comes from the lack of accuracy of coarse grid smoothers. For heavier calculations, the asymptotic convergence speed can be a more important factor and the two approaches may have comparable efficiencies.

5.3.4. Application to a locally refined mesh

The combination of a local mesh refinement with a multigrid algorithm is frequently advocated: successive grid levels are constructed by **local** refinement of the previous grid level. One disadvantage of that approach is that these levels operate only **locally** and this may reduce the speed-up(that is the ratio of efficiency between 1-grid and multigrid algorithms) with respect to the standard global multigridding, since in some region only the 1-G scheme is applied. Another approach is to vary the number of grids in order to take into account the refinement [7].

In this section, we wish to demonstrate that the coarsening / agglomerating algorithm enables us to generate **global** coarse levels, in order to keep the complete multigrid speed-up. The point of view of recomputing new coarsening after an local enrichment phase is efficient because the coarsening phase is not much CPU consuming.

We start from a locally refined mesh, constructed for the calculation of the flow past a cylinder. The fine mesh contains 2141 nodes ; then a medium mesh is derived, containing 598 zones, and finally a coarse one with 244 zones. The ratio of the levels is satisfactory. In Fig. 10a,b,c,d, the different levels are shown to demonstrate the regularity of the partitions. To compare algorithms we considered the case of a freestream Mach number of 0.38. It appears from Fig. 10e. that applying the second-order method

(over the fine level only) compared to applying the first-order method over all levels results in a reduction of convergence rate (in terms of iterations) by a factor noticeably less than 2.

5.3.5. 3-D calculations

A few 3-D experiments have been performed. Firstly, we compute the flow in a channel with 4.2 % thick bump with a tetrahedrization which is essentially a 72x21x3 mesh (Fig. 11a). The Mach at infinity is .85. The first-order accurate mesh is used. We present in Fig. 11b. the history of the convergence when a full four-grid algorithm is applied ; it can be (favourably) compared to the 2-D 3-grid multi-triangulation calculation in [19], see Fig. 11c.

Two-jet flow in a chamber : a practical illustration of the efficiency of the 3-D code is the calculation of two impinging supersonic jets in a rocket combustion chamber. The considered mesh involves 21000 nodes. If the explicit first-order one-grid scheme is applied, starting from a uniform condition in the chamber, the convergence to steady-state is prohibitively low. On the contrary, the convergence in Full Multi-Grid mode is very fast (Fig. 12). However, the problem was too stiff for computing with MG the second-order accurate solution (maximum Mach number is about 5).

Flow past a delta wing : the flow past a delta wing with Mach number at infinity equal to 1.2 and angle of attack of 10 degrees is considered. The second-order accurate solution is computed first on a regular mesh; then the mesh is locally enriched and the solution interpolated to the new mesh; the resulting mesh contains 20.000 nodes and is presented in Fig. 13a. The convergence is rather slow : 60 cycles for 2 orders of magnitude (Fig. 13b) ; but this is enough to get a acceptable solution (Fig. 13c to 13e).

5.4. Provisional conclusion

The explicit MG scheme is an interesting alternative to explicit time stepping : the gain in efficiency for 1000 - 2000 nodes in 2-D is around 3-5 ; this is less than the gain obtained with the implicit scheme described below, for which the factor is generally better than 4; one question arises : could these two approaches efficiently combine ?

6. IMPLICIT MG SCHEME

6.1. Motivation of the Correction Scheme option

For the combination of the implicit and MG ideas, one attractive choice is the derivation of a FAS nonlinear implicit scheme; such algorithms proved (within structured meshes) to be very efficient [24], [26].

However, in this study, we chose include a MG algorithm in the linear phase of the unfactored linearized implicit 2-D scheme described in Section 2. This "Correction Scheme" (CS) has been already studied by Mulder [27] and Haenel and al [29]. In choosing this option, we want firstly to upgrade the implicit code presented in Section 2; and secondly, to get informations related to the linear case before studying the nonlinear one. Furthermore, it must be noted that such an approach combines two algorithms whose convergence rates are essentially mesh independent, namely the implicit formulation as indicated in Section 2, and the MG iteration. Lastly, we wish to point out two important domains of potential good application of this approach:

-For unsteady calculations of low Mach number flows, the interest of linearized implicit formulation has been stressed in [20], [28], although the linear system is stiff and rather difficult to solve with simple iterative solution algorithms.

-For (steady) high Mach flows, usual nonlinear FAS-MG schemes, that generally do not satisfy the maximum principle, are difficult to apply. Conversely the linearized implicit scheme can satisfy the maximum principle (or positiveness preservation), allowing a nonlinearly stable time-advancing.

6.2. Description of the scheme

A physical phase and a mathematical one are applied as in (13),(14); the difference is that the linear system is solved by a MG scheme; for this, the basic iteration could be a Gauss-Seidel one that can provide efficiency (via fast convergence) and robustness (lower dependency w.r.t. aligment in MG formulation); we refer to [27] for a Gauss-Seidel CS approach. In this work we preferred a Jacobi iteration because of

the advantages found in the one-grid version (low-storage and vectorisation). Further, in some case the Jacobi iteration is replaced by a RK-4 Jacobi iteration defined as follow :

$$
\begin{aligned}
X^{(0)} &= X^{iter} \\
for\ l &= 1, L \\
X^{(l)} &= X^{(0)} + \omega\alpha_l (D^n)^{-1}(\delta\hat{w} - A^n X^{(l-1)}) \\
X^{iter+1} &= X^{(L)}
\end{aligned}
$$

in which ω is a parameter having a role analog to a CFL number. For α_l chosen as below (12), ω can be efficiently chosen as large as 2.8 or 2.9.

The residuals are transferred by same transfer operators as for the explicit scheme ; but now the usual V-cycle (instead of the saw-tooth V-cycle) or a W-cycle are applied.

6.3. Numerical experiments.

The challenge is a further improvement with respect to a one-grid implicit scheme that is already fastly convergent and rather efficient: for usual meshes and transonic flows, the linear Jacobi iteration often converges with ratios about .8 ; further, only a few sweeps are sufficient for a large part of the calculation since large time-steps cannot be applied when the main features (and especially shocks) are changing in the flow variables; but this problem partly disappears when solution are computed and interpolated through a sequence of meshes with increasing fineness: then, the Newton-like iteration (in the sense of large time-steps) can be applied earlier, which gives a chance to the CS-MG approach to bring a significant acceleration.

6.3.1. Transonic flows

For a more close relation to theoretically predicted behavior, we next consider the ideal 2-Grid scheme that consists in a 2-Grid scheme in which full convergence is realized on the coarse grid. Considering only the linear convergence in the implicit phase of the first time-step, we note that mesh independency is rather well satisfied (Fig.14b) (the reduction factor is about .7). This property is unfortunately not as well satisfied in the global nonlinear process in the case of the first order accurate spatial scheme (Fig.14c). We conjecture that for heavier meshes, the nonlinear time-stepping would behave better so that a still larger efficiency could be attained.

We next present one spatially, second-order accurate calculation in order to verify the theoretical results of Desideri referred below; the calculations are those of the internal transonic flow in a channel over a 4.2 % thick bump, a GAMM Workshop test case. The freestream Mach number is 0.85. The timestep Δt is increased as the convergence proceeds according to the following empirical rules giving the C.F.L. number:

$$C.F.L. = \max\left(10^3, 10/RES\right)$$

in which "RES" denotes the L2-norm of the explicit right-hand side normalized to the initial value and n the iteration counter. The inversion of the system is made at each iteration by a 5-level multigrid algorithm. The convergence history is shown on Figure 15, where it appears that after some initial phase during which transient phenomena are observed, the .5 asymptotic rate is achieved by the iterative method.

Consequently, in these computations, the required number of timesteps for the residual to be reduced by a prescribed factor is independent of meshsize (provided the timestep can be made large enough stably); since, in addition, the computational work involved in each inversion of the implicit system by multigrid is essentially proportional to the meshsize, so is also the computational work of the entire calculation.

An attempt to find the most efficient strategy is illustrated in Fig. 16 in which different numbers of linear cycles are applied for each time step (first-order accurate scheme) ; for 2 cycles, the multigrid scheme is about 2 times more efficient than the 1-grid one.

6.3.2. Low Mach Flow

We just present a typical result obtained on a stiff test case: in a closed rectangular vessel we consider an initial state with uniform flow at Mach Number 0.001. After a long time, the gas will be near steady

state, with zero velocity. Because of the high ratio between the different waves (acoustics, convection, ...), an explicit scheme cannot be used; we apply the implicit one with a large CFL number at 10^5 and examine the convergence of the linear system iterative solution for advancing the first time step.

It is clear that the one grid jacobi iteration is not efficient enough, due to its *explicit-like* behaviour with allows waves to travel and reflect (Fig. 17). We just mention that the Gauss-Seidel iteration is at least four times more efficient on a scalar computer; but this advantage disappears on a vector computer since only Jacobi fully vectorizes. We now apply the MG-RK4-Jacobi scheme with four grids; the improvement (Fig. 17) is striking: the convergence is 200 times faster and the efficiency is 20 times better for the linear solution ; some slowering is observed after a 10^5 residual reduction ; this phenomenon is related to ill-conditioning and disappears for lower CFL numbers (Fig. 19).

6.3.3. High Mach flows

In this study, we restrict ourselve to first order accurate calculation. This kind of flow is nonlinearly stiff and carries a lot of problems ; for a class of algorithm applying to it, see [30].

Pathological behaviour

The first test case choosen is the calculation of a Mach number 8 flow around a NACA0012 airfoil with a 20 degrees angle of attack. A 800 node mesh is first considered (as in Fig. 8).

A first series of experiments shows that the implicit schemes present a pathological behaviour already in the 1-grid case with RK1-Jacobi ; this is illustrated in two figures:

- in Fig 20, the nonlinear (time advancing) convergence is not obtained, but, instead, a limit cycle takes place.
- it is interesting to examine whether the corresponding linear phase are well converging:

We draw the following curves; each point corresponds to a time level W^{n+1} ;

-the ordinate is the corresponding nonlinear residual (norm of $\delta \bar{W}^n/\delta t$, following (13)),

-the abscissa is the **mean** residual ratio of the linear MG cycles that have been performed to derive δW^{n+1} from $\delta \bar{W}^n$;. if it is less than one, this means that a more or less good linear convergence is obtained.

Looking at Figure 21, we observe that the linear convergence failed in some time-steps, leading to some cycling between the convergence of the linear phase and the lack of it.

This problem can be related to the question of diagonal dominance in upwind matrices for nonlinear systems: this property is true when the coefficients, and then the local diagonalisation matrices, are constant ones; diagonal dominance can disappear when the flow properties present large variations; also some stagnation zones may induce very small diagonal values. We then try the Flux Vector Splitting of van Leer; the result is slightly improved, that is with a smaller amplitude in the limit cycling (Figs. 20, 21).

As a third splitting that involves a more isotropic non vanishing numerical viscosity, we introduce the following one (define in the 1-D case):

$$\Phi(U,V) = (\frac{1}{2})\,(F(U)+F(V)) - (\frac{1}{2})\,T|\Lambda^*|T^{-1}(V-U) \tag{15}$$

with

$$T = T(\frac{(U+V)}{2})$$

$$\Lambda = \Lambda(\frac{(U+V)}{2}) = Diag(\lambda_k)$$

$$|\Lambda^*| = Diag(max(|\lambda_k|, c))$$

where c is the sound speed. This splitting is introduced only in the implicit matrix while the Osher splitting is maintained in the explicit part. Since this splitting is less consistent with the Osher's than Steger's is, the nonlinear convergence in the one-grid version is generally rather slowened (a factor of

about 30 purcent) ; on the contrary, for this pathological case, the new preconditioning makes the limit cycling disappear, as depicted in Figs. 20, 21.

The multigrid RK-4 jacobi scheme is again applied, with a so-called W^* cycle, a W-cycle in with smoothing is not applied before going to upper level, this reduces cost to 2 work units, that is 8 fluxes.

We observe that the linear phase of the implicit scheme is much better solved ; however this results in only a slight improvement the nonlinear convergence with respect to the one-grid scheme. Turning to a 3000-node mesh, we get a gain in efficiency (scalar computing) of 1.6 w.r.t. a one grid calculation, both calculations being initialized by a coarse mesh (800 nodes) solution; the convergence comparison is presented in Figs. 22, 23.

Blunt body flow

A set of experiments with an hypersonic workshop test case is also presented ; it is a Mach number 8.15 flow around a double ellipse ; the flux (15) is applied in both implicit and explicit phases the CFL number ; we compare the convergence on two successive meshes :

(a) on a 1700 node mesh (Fig. 24), convergence is presented after the first "research phase" (after residual division by 10.000) in which the convergence is slow due to the stiffness of the evolution starting from the uniform flow. The number of linear cycles is 45 for 4 orders of further residual reduction (Figs. 25, 26).

(b) then the solution is interpolated to a 6000 nodes mesh (Fig. 27) ; the convergence is then really comparable to the 1700-node one (Figs. 28, 29) ; the solution is presented in Fig. 30. The number of linear cycles is 68. The gain of efficiency is about 2. For the convergence final phase, the number of linear cycles is about the same for the same convergence rate.

7. CONCLUSION

In this paper, we have presented a new method for automated coarsening in order to compute steady Euler flows with non-structured meshes. This method is introduced into two MG schemes, an explicit FAS one, and an implicit CS one. We show that the method behaves like a standard one. In an ideal situation, the resulting schemes should be infinitely more efficient than one-grid ones when applied to infinitely large meshes; a series of experiments is described in order to measure the gain for a sample of representative meshes.

The explicit FAS-MG scheme brought some noticeable improvements with respect to the corresponding explicit one-grid scheme, with a CPU gain between 3 and 5, although we noted that the behavior with second-order accuracy is less good than with first-order (less mesh independance); we refer to [25] for further improvements.

However, many stiff cases (high Mach) could not be treated because of spatial instabilities negative pressure values.

To escape this trouble, we considered an implicit scheme : the implicit correction scheme brings robustness and permits some improvements with respect to the implicit one-grid scheme in particular :

For low Mach number, unsteady, flows a rather large gain (about 20) is obtained.

For high Mach flows, the monotonicity problem is dealt with by increasing progressively the time-step; the overall gain is between 1.5 and 2 for rather fine meshes with respect to the implicit scheme, between 5 and 8 w.r.t. the explicit one. However, some pathology had to be treated, and the acceleration in linear phase is difficult to transform in global acceleration.

The whole present work has been restricted to fully vectorizable and parallelizable (explicit-like) algorithms, and brought some hints about the behaviour of this class of schemes.

8. ACKNOWLEDGEMENTS

We thank L. Fezoui for her contribution at different stages of this study. We thank J.-A. Desideri, E. Hettena for yielding the double ellipse mesh and the Simulog team for yielding the 3-D chamber mesh.

This study was partly supported by Group 6 of DRET ; M-H. Lallemand was also partly supported by an INRIA-CNES fellowship grant.

REFERENCES

[1] ANGRAND F. - LEYLAND P., *Schéma multigrille dynamique pour la simulation d'écoulements de fluides visqueux compressibles*, INRIA Research report 659 (1987).

[2] BANK R. - SHERMAN A., *A multi-level iterative method for solving finite-element equations*, Proceedings of the fifth symposium on reservoir simulation, pp. 117-126, Society of petroleum engineers of AIME, (1980).

[3] PEREZ E. - PERIAUX J. - ROSENBLUM J-P. - STOUFFLET B. - DERVIEUX A. - LALLEMAND M-H., *Adaptive full-multigrid finite element methods for solving the two-dimensional Euler equations*, I.C. 10 NMFD, Pekin (1986), Lecture Notes in Physics no 254, Springer Verlag.

[4] PEREZ E., *Finite element and multigrid solution of the two- dimensional Euler equations on a non-structured mesh*, INRIA Research Report 442 (1985).

[5] LALLEMAND M-H. - FEZOUI L. - PEREZ E., *Un schema multigrille en éléments finis décentrés pour les equations d'Euler*, INRIA Research Report 602 (1987).

[6] LOHNER R. - MORGAN K., *Unstructured Multigrid methods*, Second European Conference on Multigrid Methods, Koln (RFA), October 1-4, 1985.

[7] MAVRIPLIS D. - JAMESON A., *A Multi-grid solution of the two-dimensional Euler equations on unstructured triangular meshes*, AIAA Paper 87-0353.

[8] LECLERCQ M-P., *Résolution des équations d'Euler par des méthodes multigrilles ; conditions aux limites en régime hypersonique*, Thesis St-Etienne (F), 1990.

[9] BRANDT A. - MC CORMICK S.F. - RUGE J., *Algebraic multigrid (AMG) for space matrix equations*, in Sparsity and its applications, (D.J. Evans Ed.), Cambridge University Press (1984).

[10] FEZOUI L., *Résolution des équations d'Euler par un schéma de Van Leer en éléments finis*, INRIA Research Report 358, (1985).

[11] FEZOUI L. - STOUFFLET B., *A class of Implicit upwind schemes for Euler simulations with unstructured meshes*, Journal of Comp. Phys., vol. 84 (1989), no 1, p. 174-206.

[12] ZIENKIEWICZ O.C., *The Finite Element in Engineering Science*, Mc Graw Hill London (1971).

[13] OSHER S. - SOLOMON F., *Upwind difference schemes for hyperbolic systems of conservation laws*, Journal Math. Computation, (1986).

[14] STOUFFLET B. - PERIAUX J. - FEZOUI F. - DERVIEUX A., *Numerical simulation of 3-D hypersonic Euler flows around spaces vehicles*, AIAA paper 87-0560 Reno, Nevada (1987).

[15] VAN LEER B., *Flux Vector Splitting for the Euler equations*, Lecture Notes in Physics, vol. 170, p. 405-512 (1982).

[16] VAN LEER B., *Computational methods for ideal compressible flow*, Von Karman Institute, Lectures Series 1983-04 Computational Fluid Dynamic (1983).

[17] JAMESON A., *Numerical solution of the Euler equations for compressible inviscid fluids*, Numerical methods for the Euler equation of fluid dynamics, F. Angrand et al. Eds., SIAM Philadelphia (1985).

[18] TURKEL E. - VAN LEER B., *Flux vector splitting and Runge-kutta methods for the Euler equations*, ICASE Report 84-27, June 1984.

[19] LALLEMAND M-H., *Schémas décentrés multigrilles pour la résolution des équations d'Euler en éléments finis*, Thesis, Univ. of Marseilles (F) 1988.

[20] BENKHALDOUN F. - DERVIEUX A. - FERNANDEZ G. - GUILLARD H. - LARROUTUROU B., *Some investigations of finite-element solutions to stiff combustion problems : mesh adaption and implicit time-stepping*, Proceedings of the NATO Advanced Research Workshop on Combustion Modelling and its Application, Brauner and Schmidt-Lainé eds., 1987.

[21] STEVE H., *Schémas implicites linéarisés décentrés pour la résolution des équations d'Euler en plusieurs dimensions*, Thesis, Marseille (F), (1988).

[22] DESIDERI J-A., *Preliminary results on iterative convergence of a class of implicit schemes*, INRIA Research Report 490 (1986).

[23] BOEHMER K. - HEMKER P. - STETTER H.J., *The Defect Correction approach*, Computing Suppl., 5, 1-32 (1984).

[24] HEMKER P. - SPEKREIJSE S., *Multiple grid and Osher's scheme for the efficient solution of the steady Euler equations*, Appl. Num. Math. 2 (1986), 475-493.

[25] KOREN B. - LALLEMAND M-H., *Iterative defect correction and multigrid accelerated explicit time stepping schemes for the steady Euler equations*, CWI Report NM-R8908 (1989).

[26] ANDERSON W.K. - THOMAS J-L. - WHITFIELD D.L., *Multigrid acceleration of the flux split Euler equations*, AIAA 86-0274 (1986).

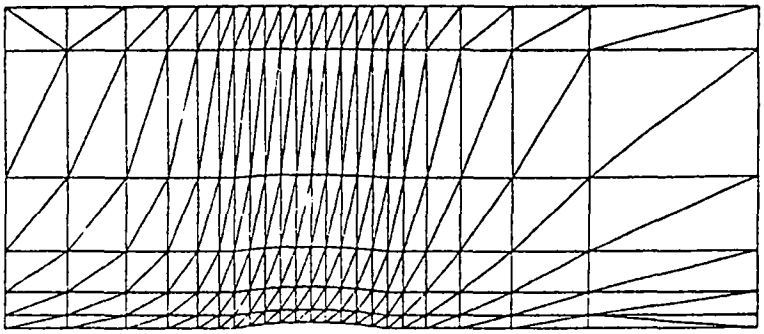
[27] MULDER W.A., *Multigrid relaxation for the Euler equations*, J. Comp. Phys. 77 (1988), 183-206.

[28] FERNANDEZ G. - GUILLARD H., *Implicit schemes for combustion problems*, Lecture Notes in Physics, 351, 277-286 (1989).

[29] HAENEL D. - MEINKE M. - SCHOEDER W., *Application of the Multigrid method in solutions of the compressible Navier-Stokes equations*, in Proc. of the 4th Copper Mountain Conf. on Multigrid Methods, SIAM proc. series, 234-254 (1989).

[30] KOREN B., *Robustness improvement of point Gauss-Seidel relaxation for steady, hypersonic flow computations*, CWI NM-N8805, Amsterdam (1988).

a



b

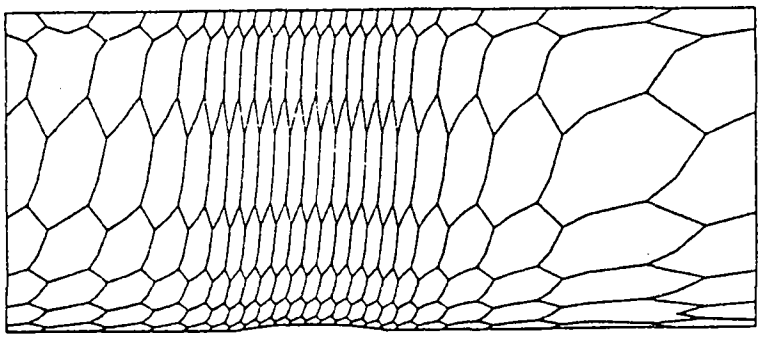
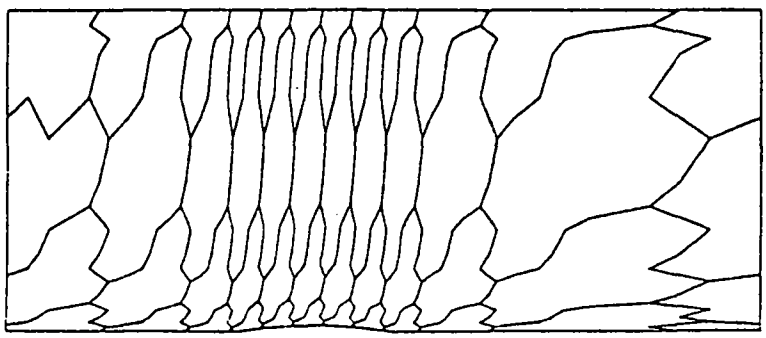


Figure 6a : Flow in a channel with a 4.2% thick circular bump.
Mach at infinity = .85 ; a : triangulation ; b : dual mesh

a



b

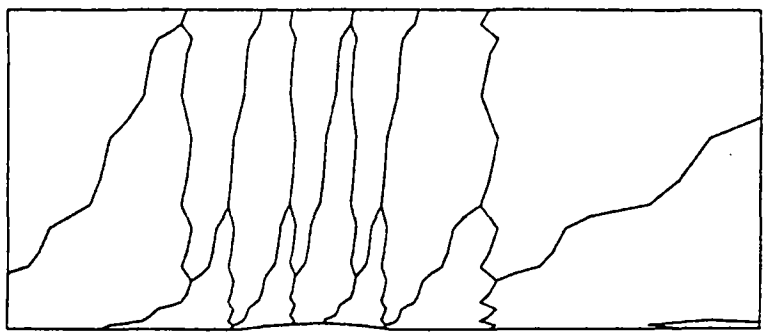


Figure 6b : Flow in a channel with a 4.2% thick circular bump.
Mach at infinity = .85 ; a : medium level ; b : coarse level

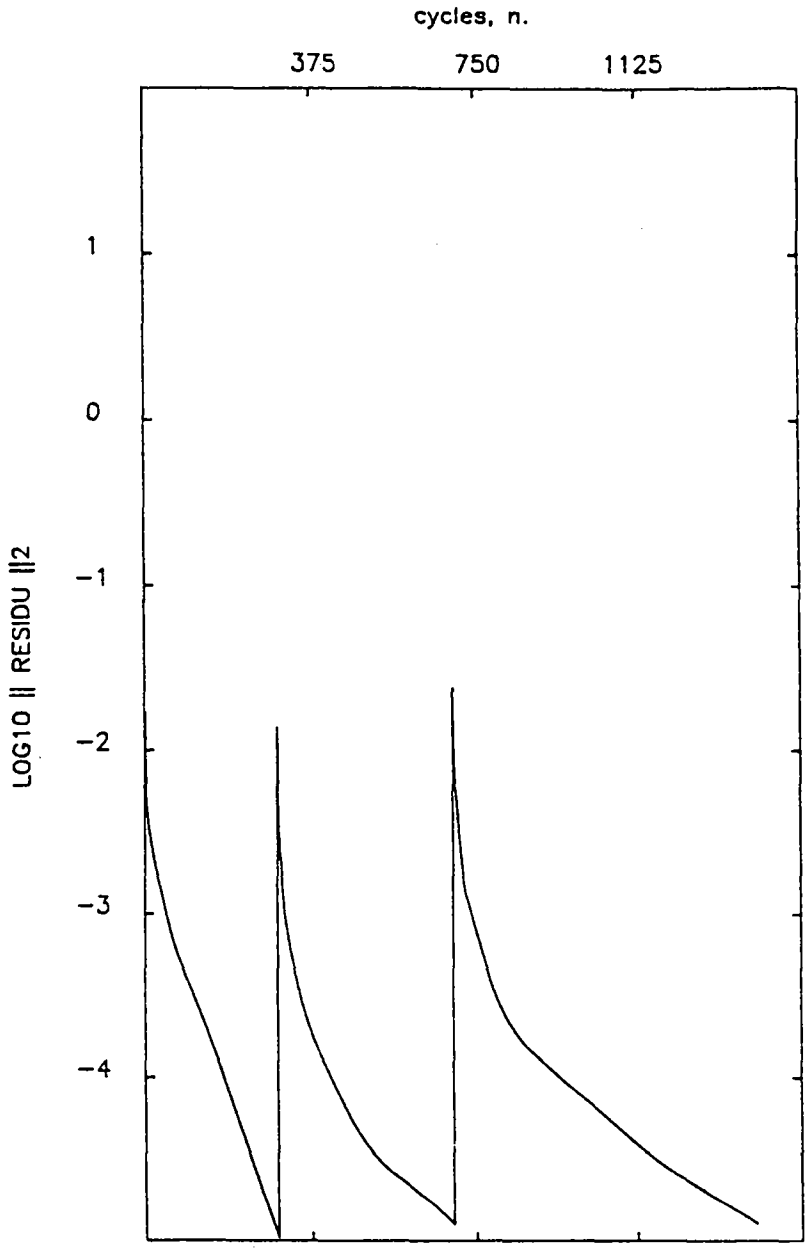


Figure 6c : Convergence histories - one-grid RK1 first-order upwind scheme with coarse level initial solution.

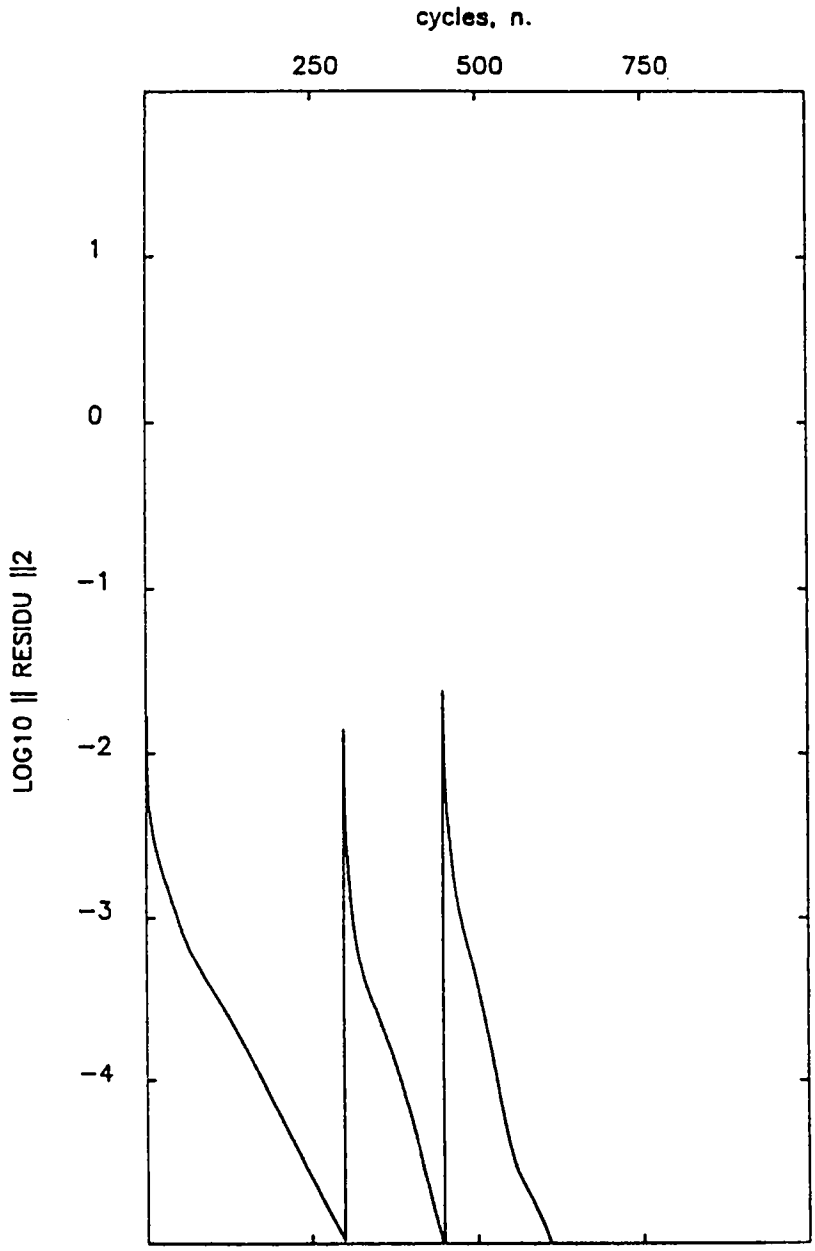
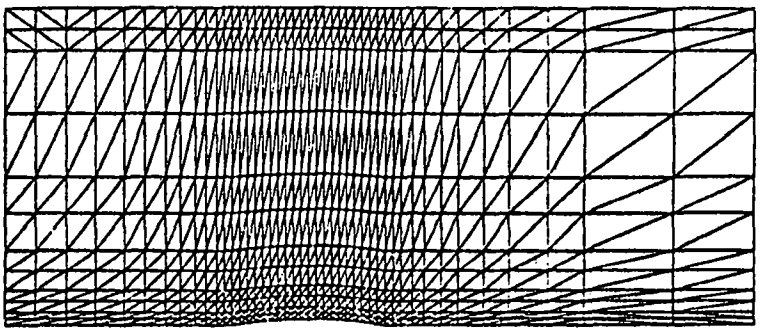


Figure 6d : Convergence histories - FMG RK1
first-order upwind scheme.

a



b

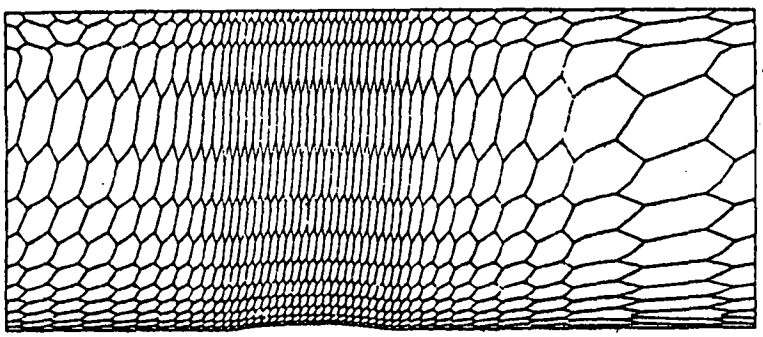
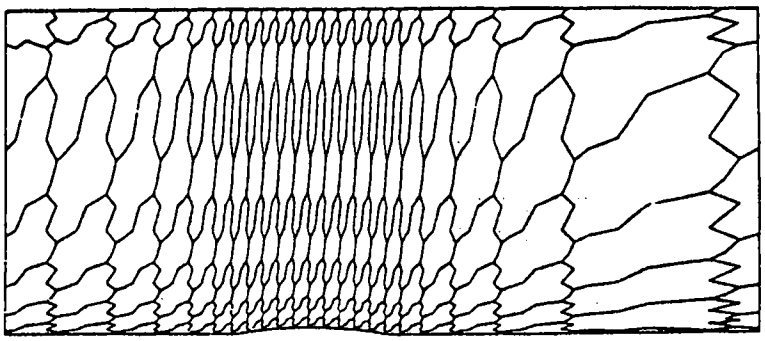
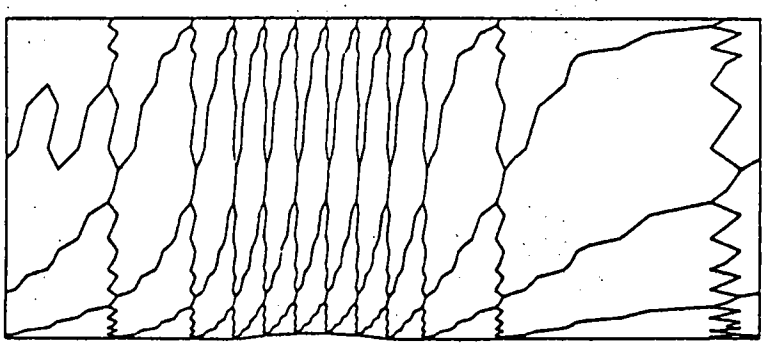


Figure 7a : Same as Figure 6a, with a finer triangulation.

a



b



c

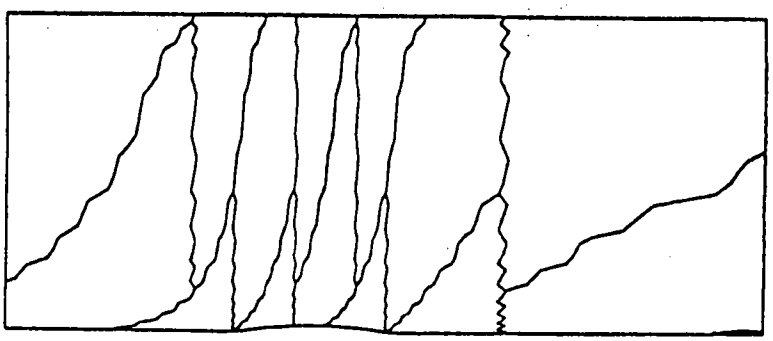


Figure 7b : Same as Figure 6b, with a finer triangulation

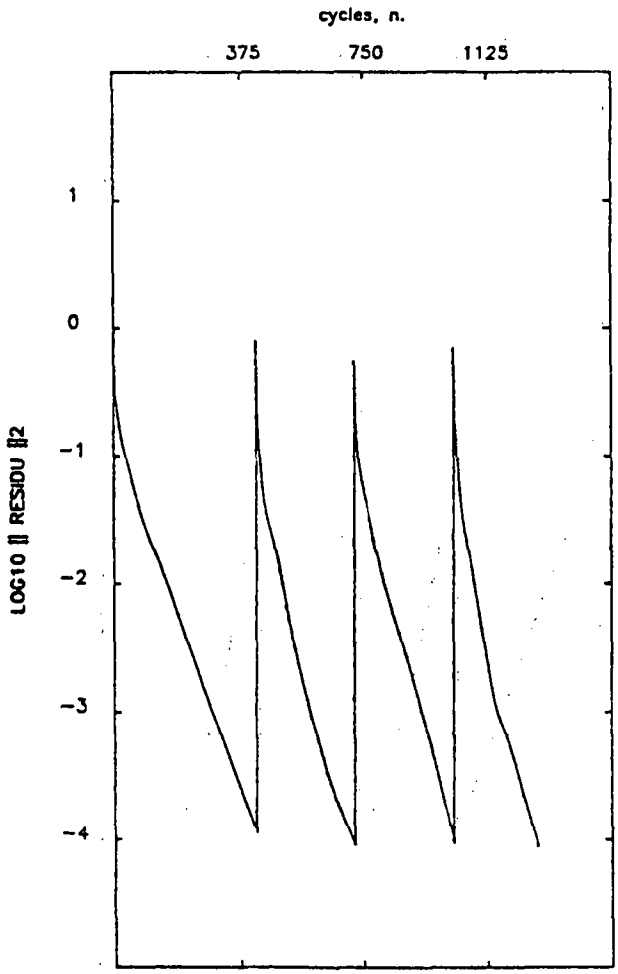
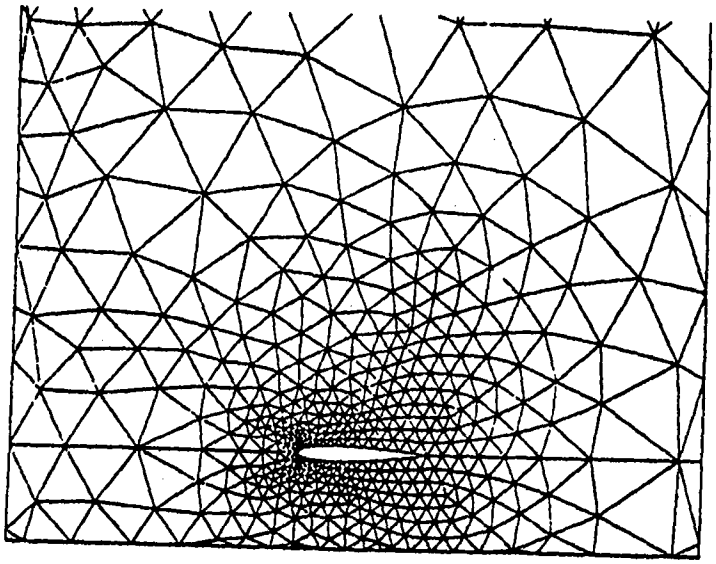


Figure 7c : Four-grid calculation : FMG convergence
(RK1 first-order upwind scheme).

a



b

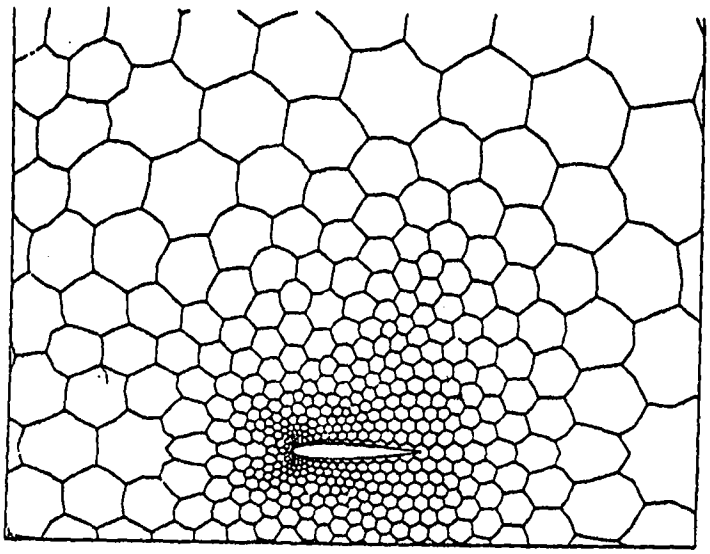
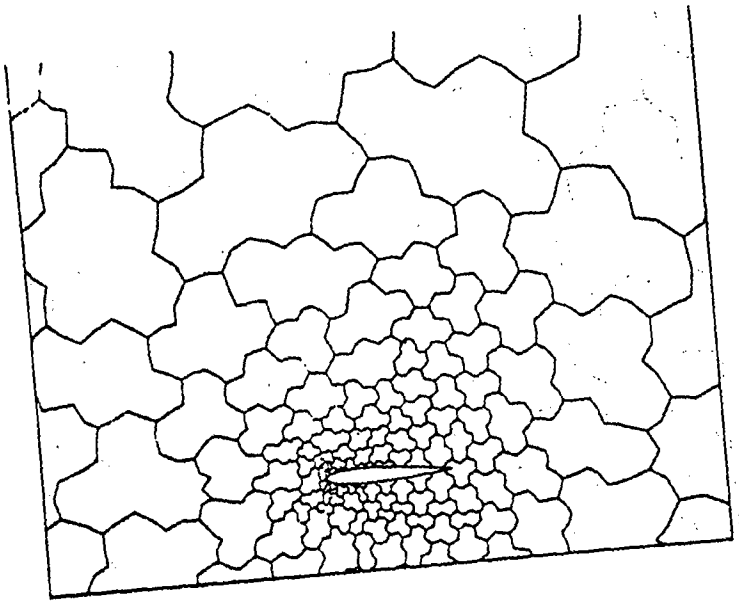


Figure 8 : Flow around a NACA0012 airfoil. Mach at infinity =0.72, angle of attack =0. deg. a : triangulation ; b : dual mesh.

c



d

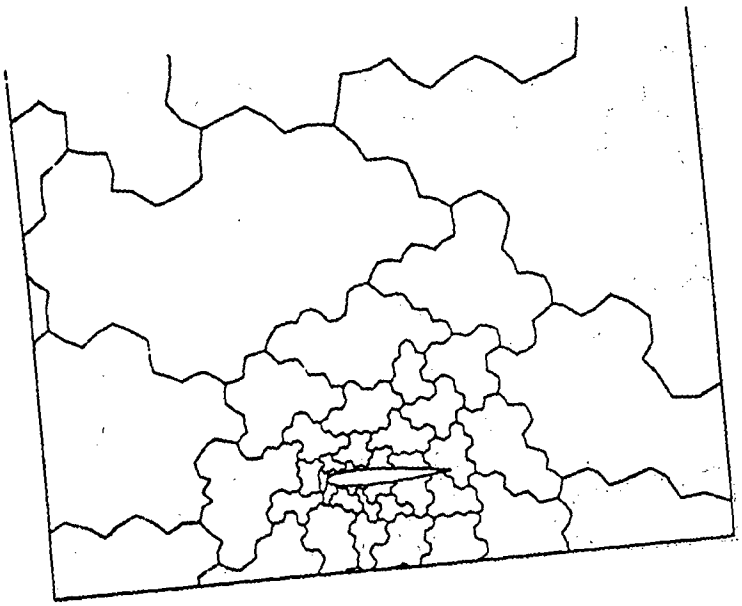


Figure 8c and 8d : Flow around a NACA0012 airfoil. Mach at infinity = 0.72, angle of attack =0. deg. Coarse meshes.

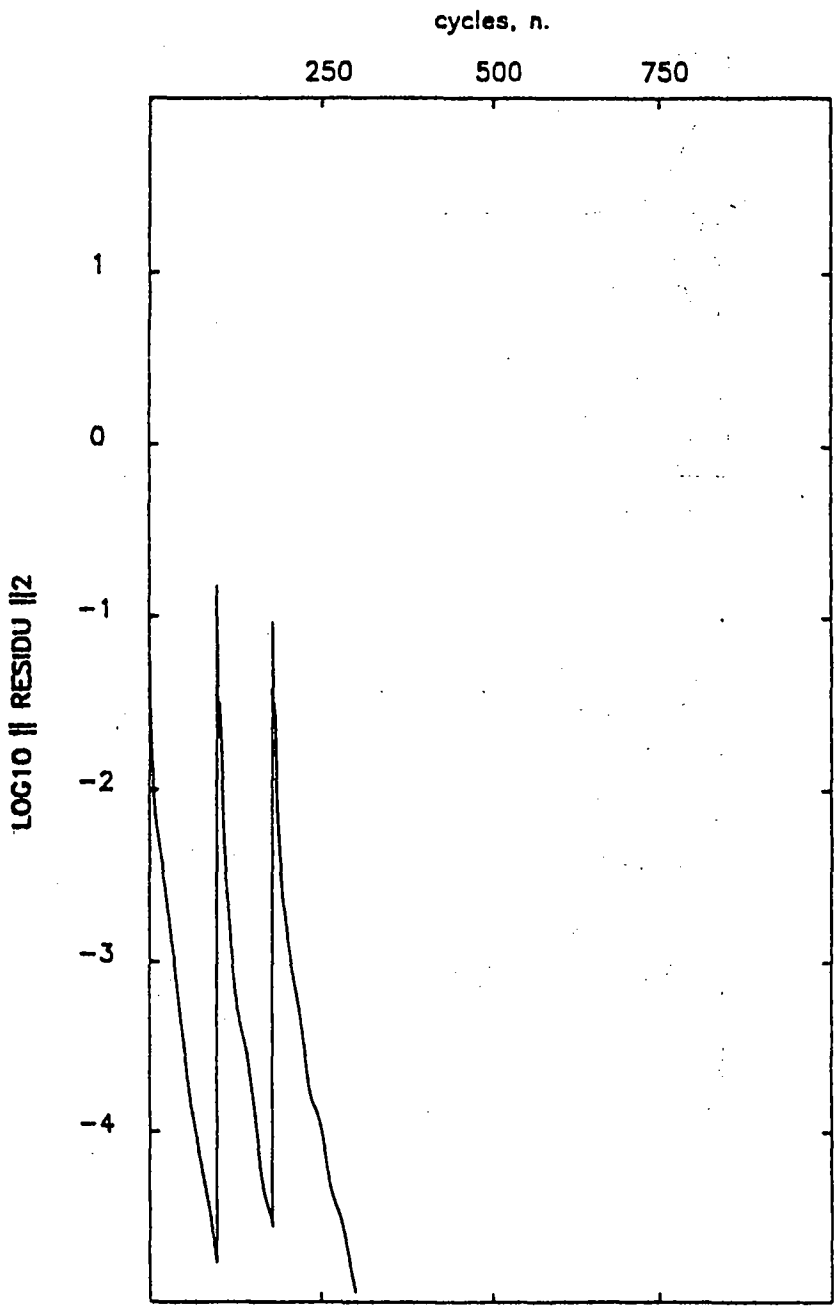


Figure 8e : FMG convergence with first-order scheme.

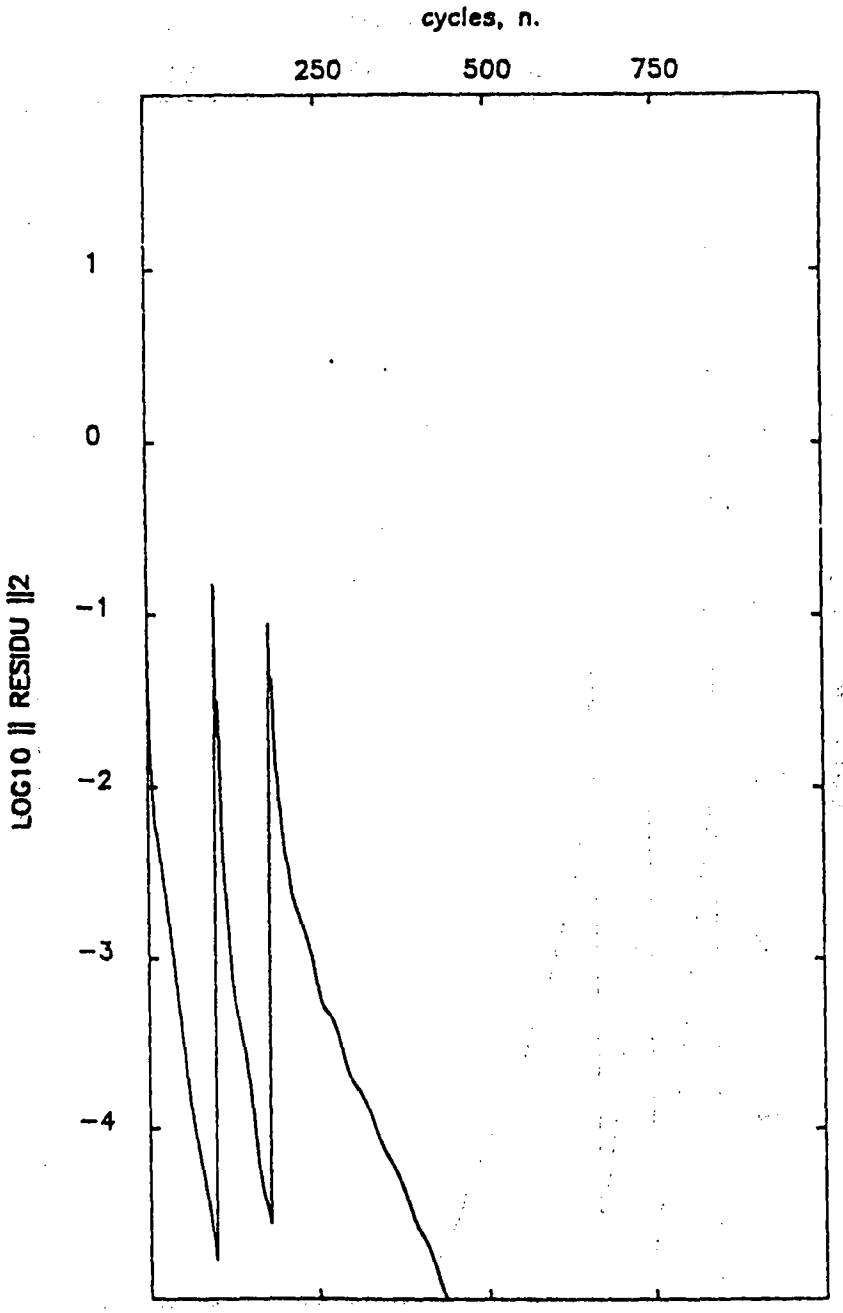


Figure 8f : FMG convergence with second-order scheme.

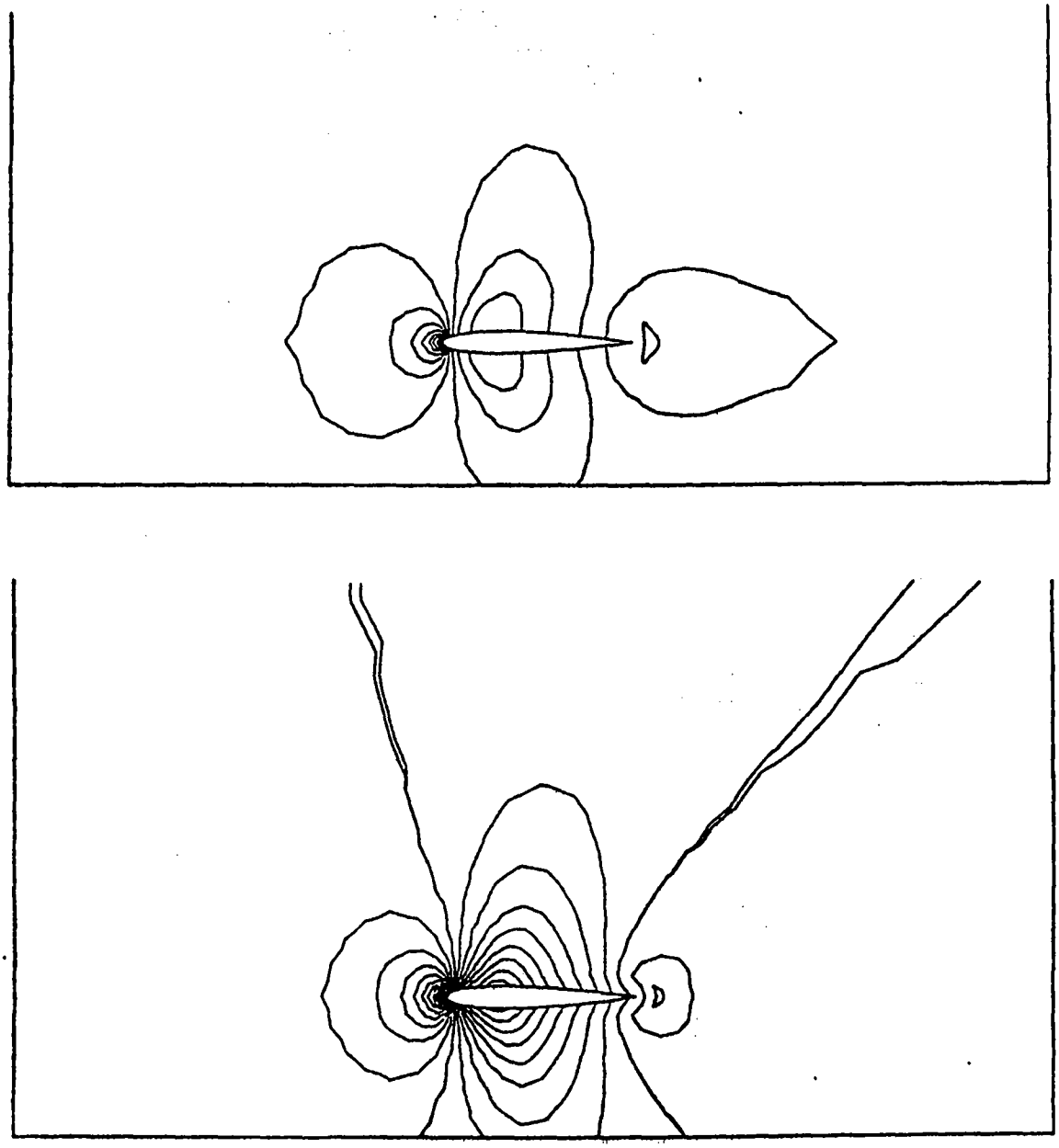
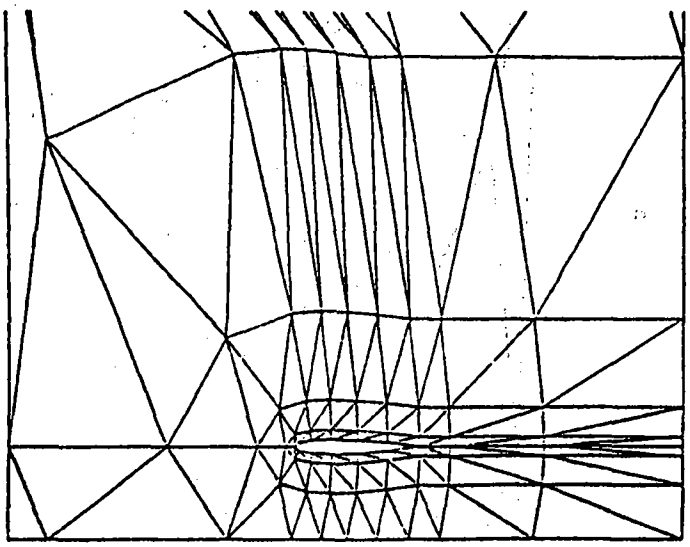


Figure 8g : Flow around a NACA0012 airfoil, continued Mach and pressure contours (Mach at infinity = 0.72.)

a



b

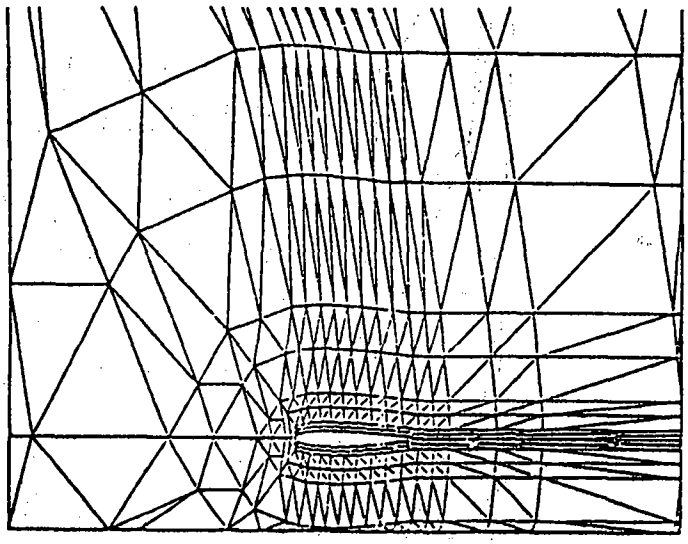
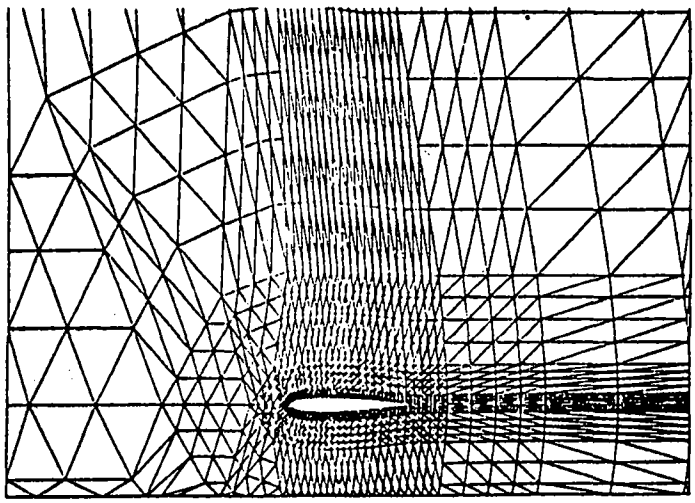


Figure 9a and 9b : The coarse triangulation used in the multi-triangulation (MT) scheme.

c



d

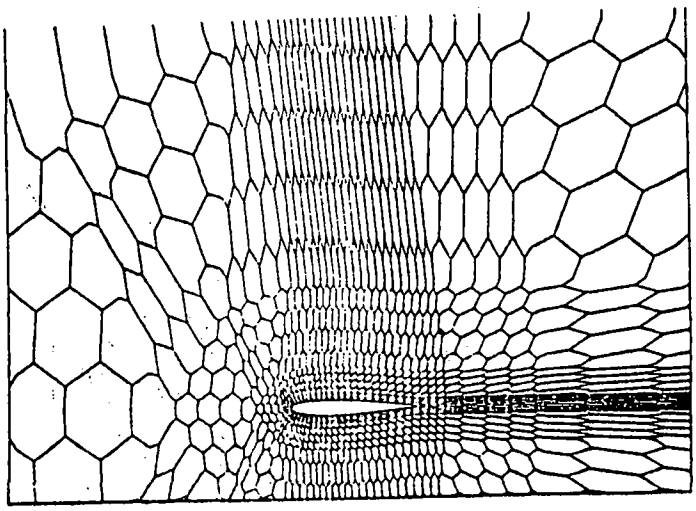
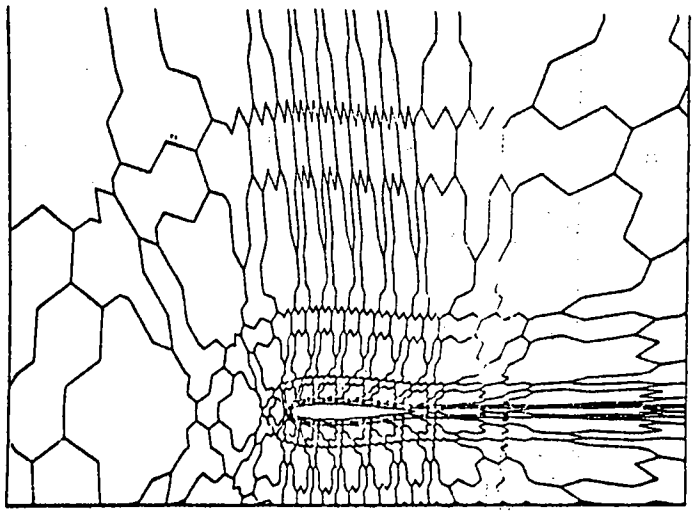


Figure 9c and 9d : The fine levels used in the MT scheme and presented scheme. c : triangulation ; d : dual fine mesh.

e



f

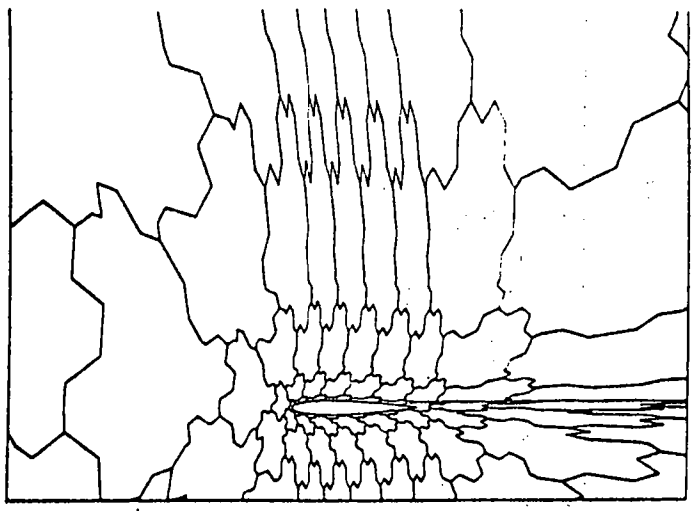
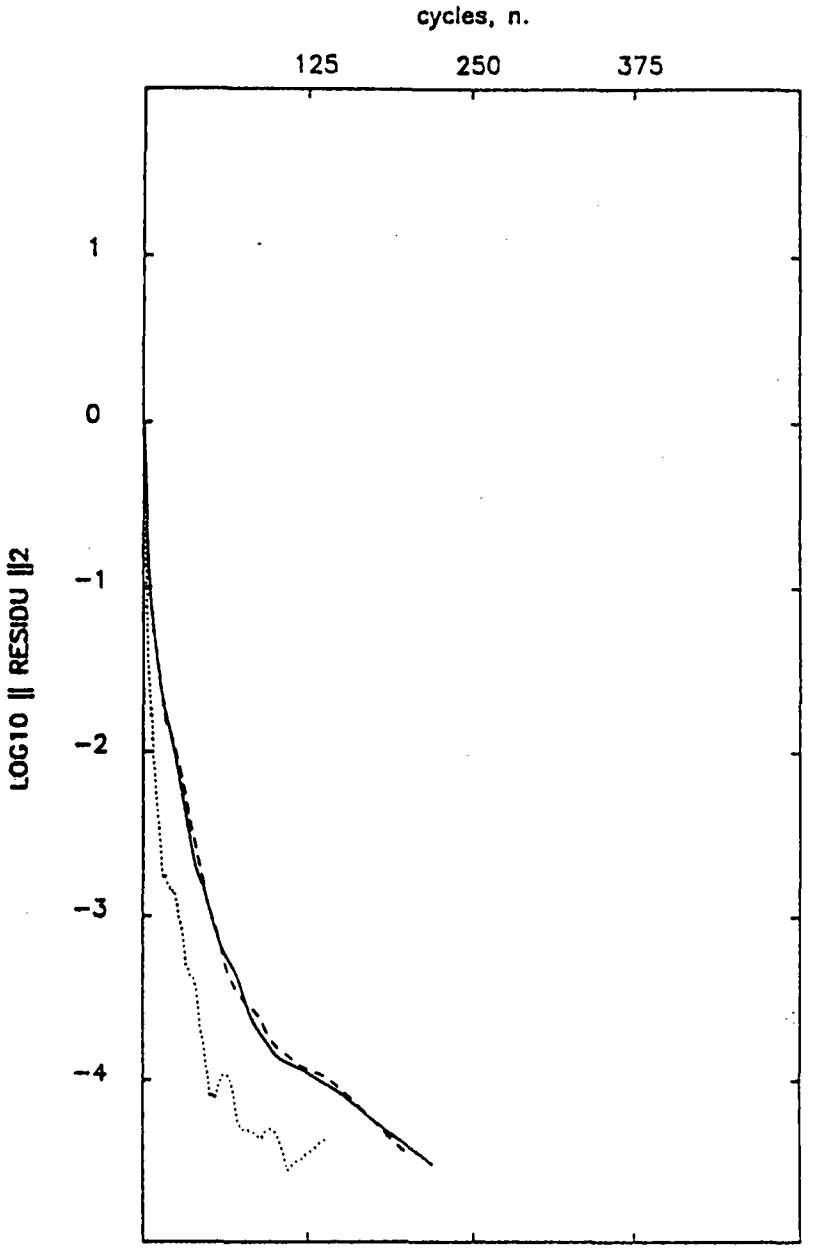


Figure 9e and 9f : The coarse levels used in the presented scheme.
e : medium level ; f : coarse level.



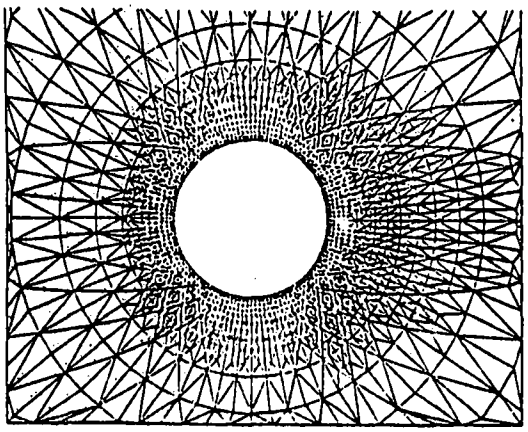
····· MT algorithm with 2nd–order scheme applied on each level.

---- MT algorithm with 2nd–order scheme applied on the fine level only.

——— presented 2nd–order version agglomerating algorithm.

Figure 9g : Flow around a NACA0012 airfoil ; comparison with a multi-triangulation (MT) scheme.

a



b

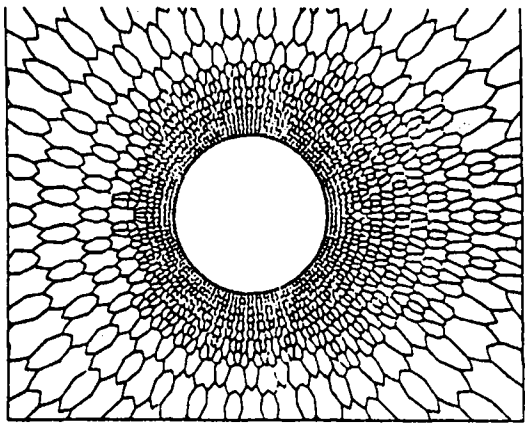
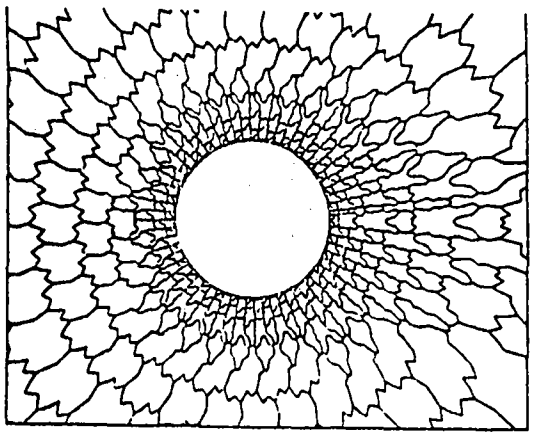


Figure 10a and 10b : Flow past a cylinder. a : fine triangulation
b : dual fine mesh.

a



b

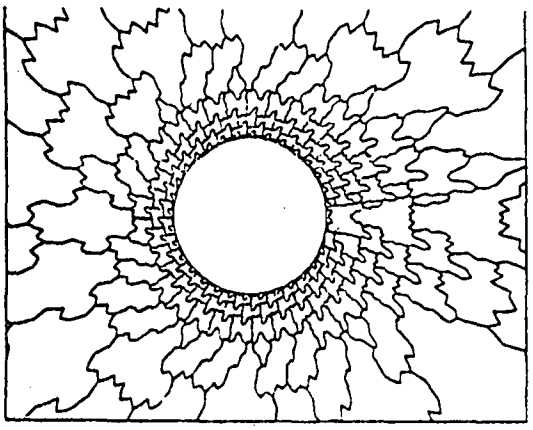
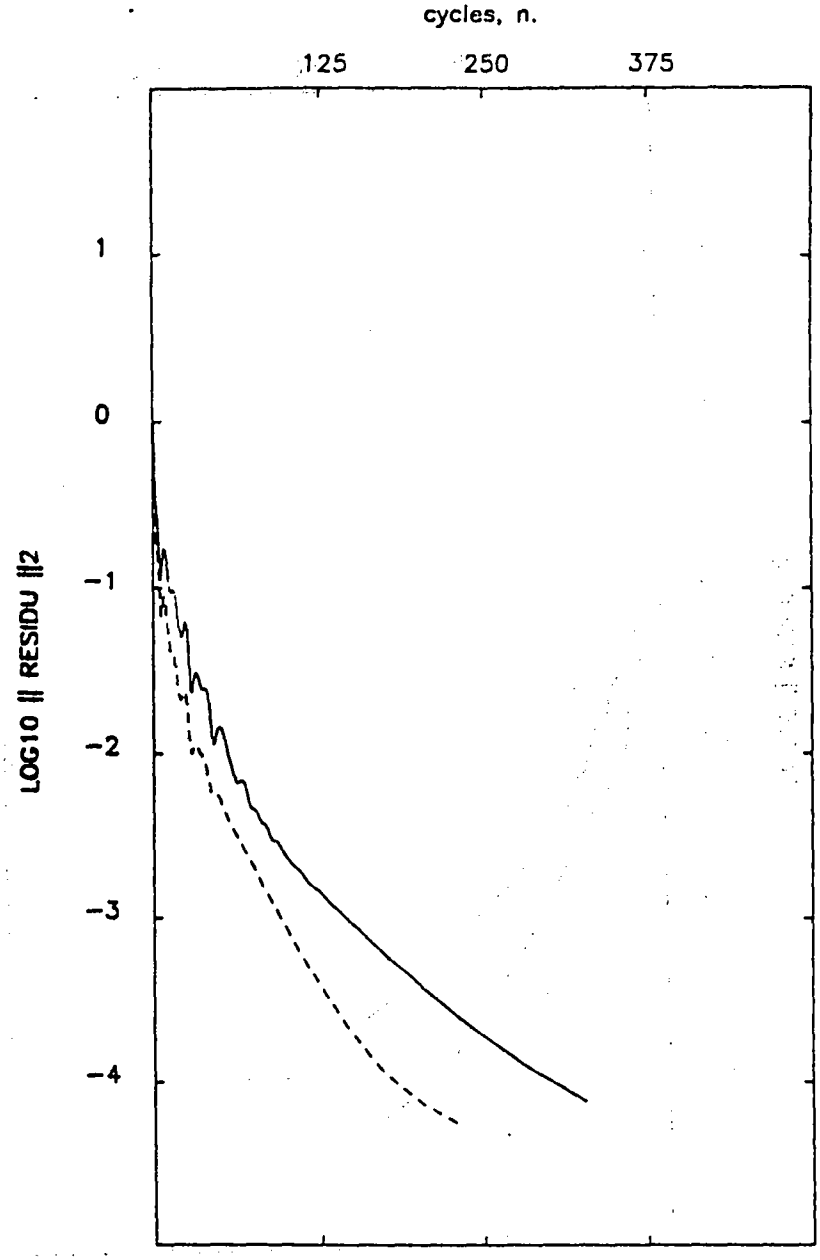


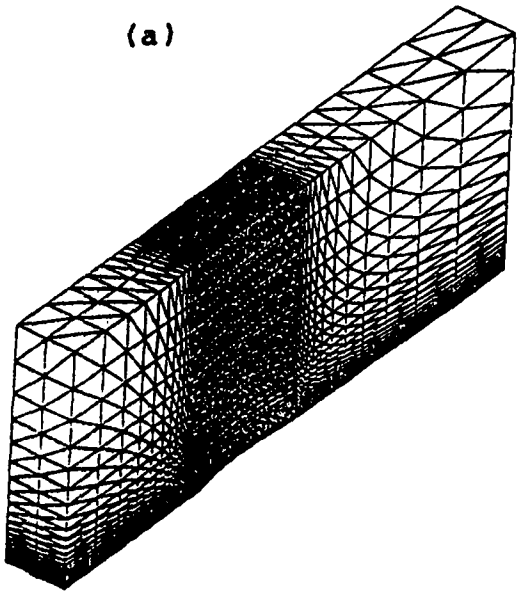
Figure 10c and 10d : Flow past a cylinder. c : medium level. d : coarse level.



—— 2^{nd}-order.

– – – 1^{st}-order.

Figure 10e : Flow past a cylinder. Mach at infinity = 0.38.
Comparison between 1st-order and 2nd-order version of the 3-grid algorithm.



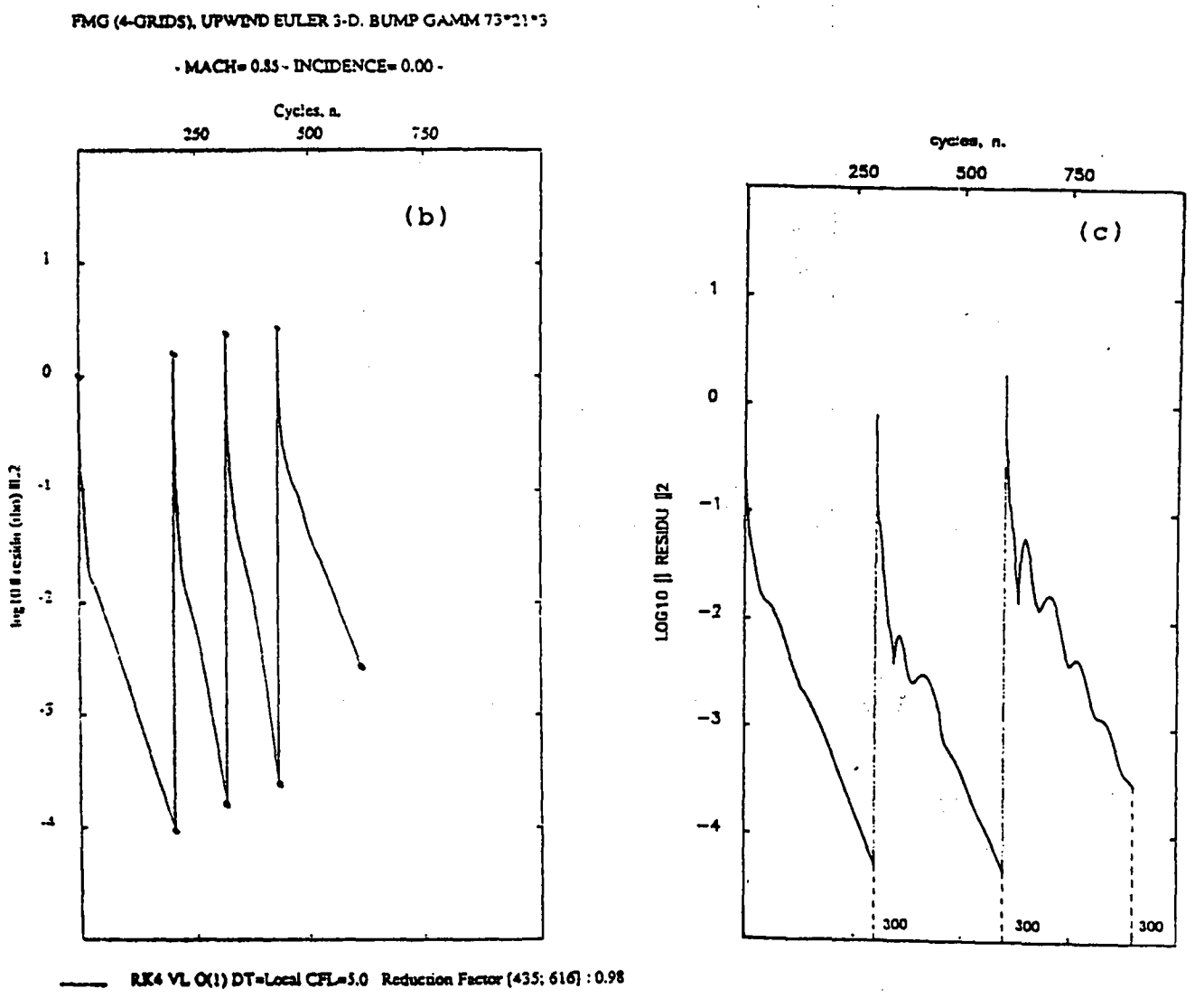
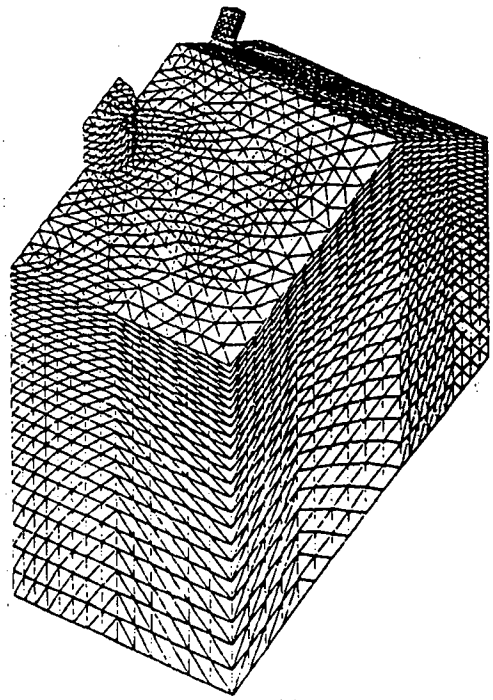
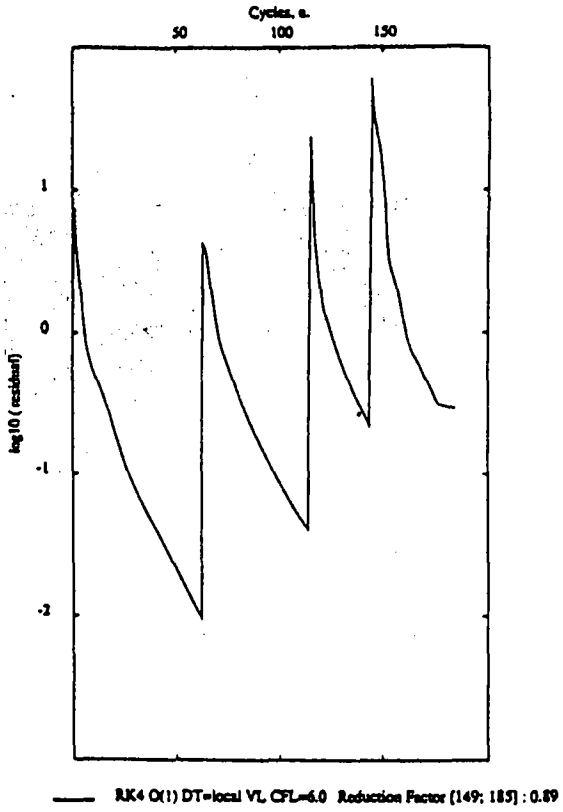
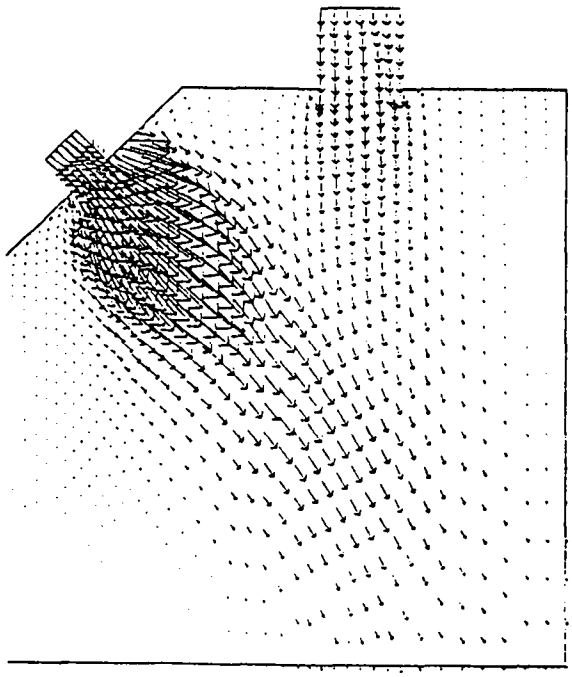


Figure 11a to 11c : First-order 3-D scheme ; a comparison of 3-D FMG (a,b) with a 2-D multitriangulation analog (c).

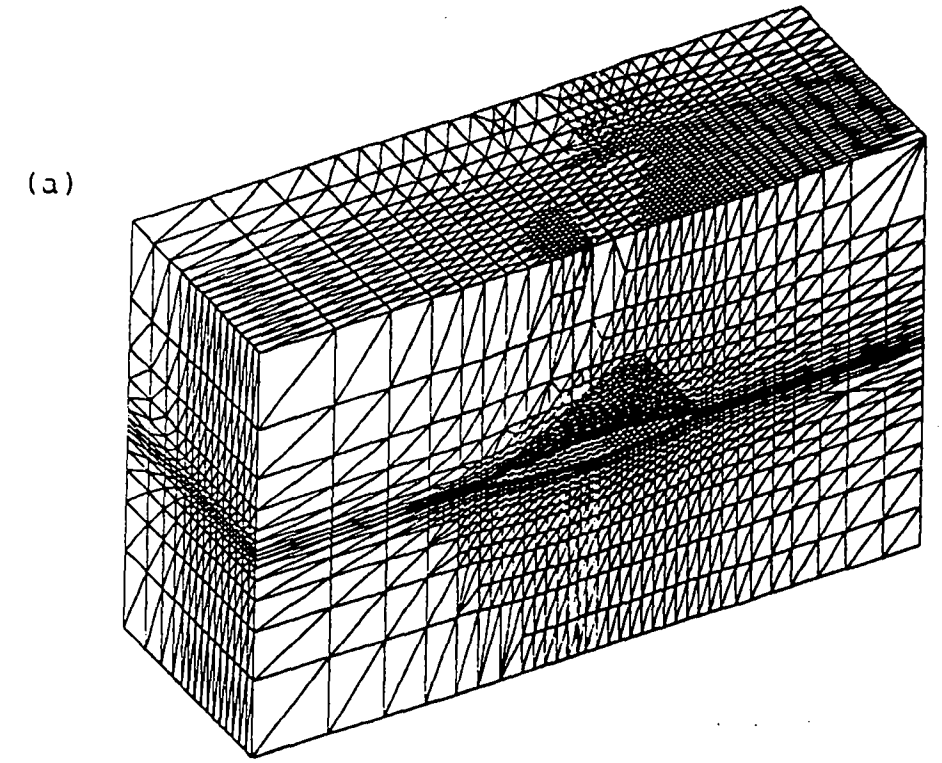






ITERATION= 170

Figure 12a to 12 c : First-order 3-D scheme : Transonic impinging jets in a combustion chamber (21 000 nodes)



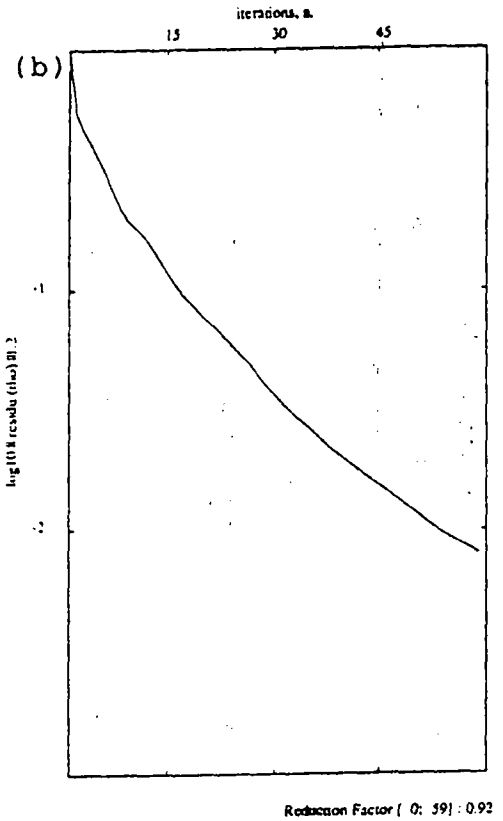
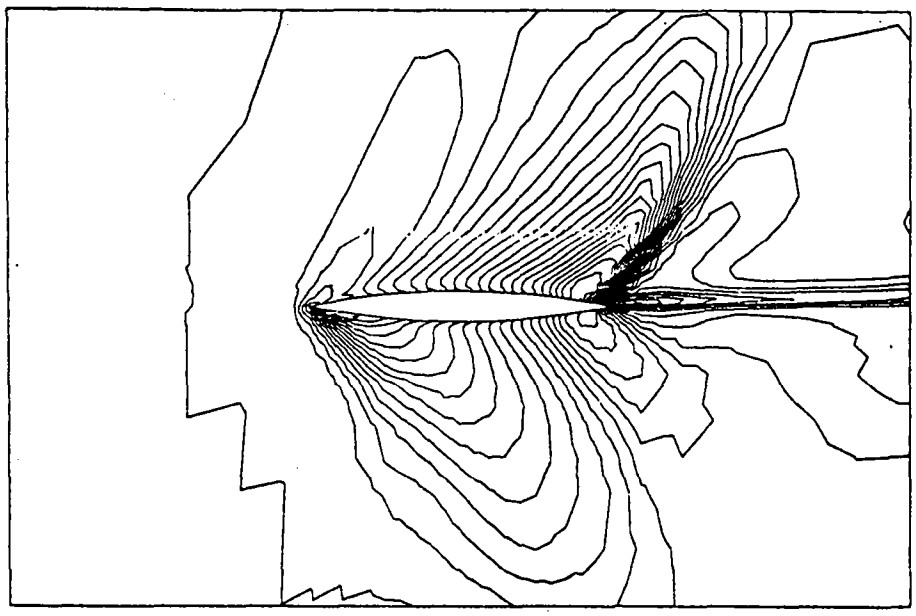
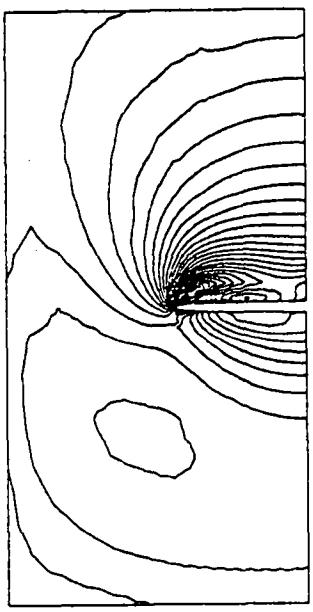


Figure 13a and 13b : Flow past a delta wing :
13a : sketch of the enriched mesh (20000 nodes)
13b : convergence history, when initial condition is the coarse mesh solution.

(c)



(d)



(e)

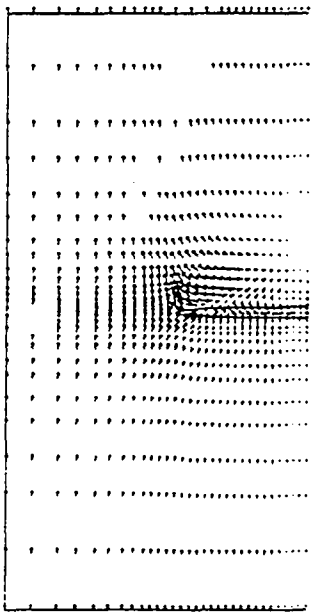
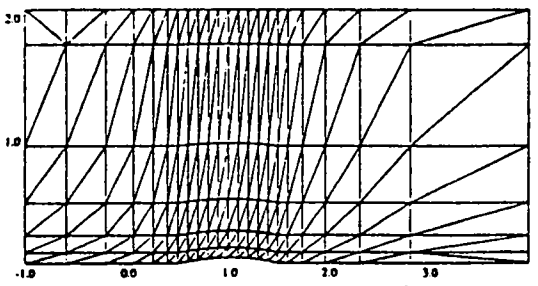
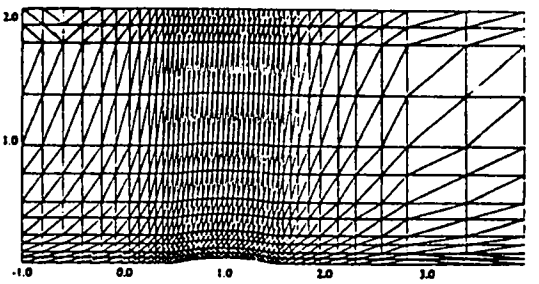


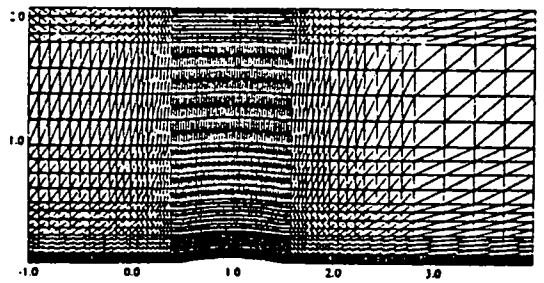
Figure 13c to 13e : Flow past a delta wing :
13c : Mach contours on the symmetry plane
13d : Mach contours on a normal plane at 90% of the chord
13e : The corresponding velocity field.



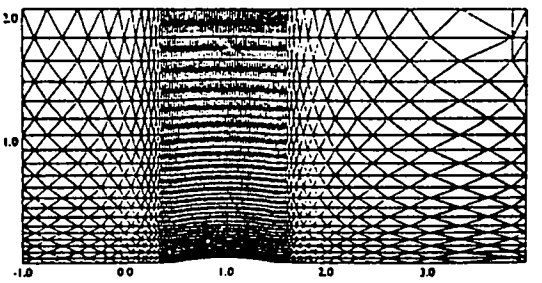
T_1 : 161 points.



T_2 : 585 points.



T_3 : 2225 points.



T_4 : 1512 points.

Figure 14a : Triangulation T_1, T_2, T_3 and T_4.

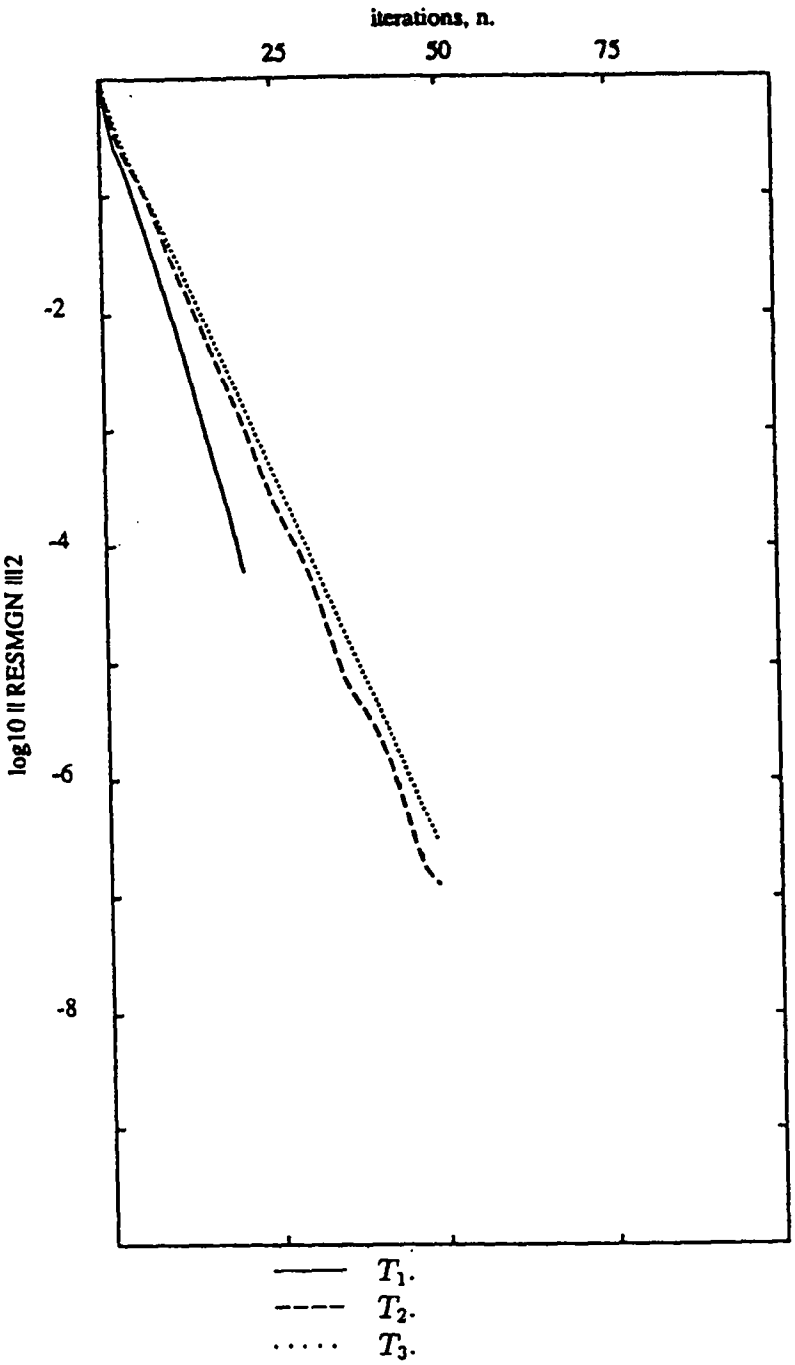


Figure 14b : Convergence in the linear phase, ideal bigrid scheme ; channel with bump ; Mach = 0.85. First-order spatial approximation First time-step, CFL = 1000. Jacobi relaxation, $\omega = 1$

2-grid with T_1
2-grid with T_2
2-grid with T_3
2-grid with T_4

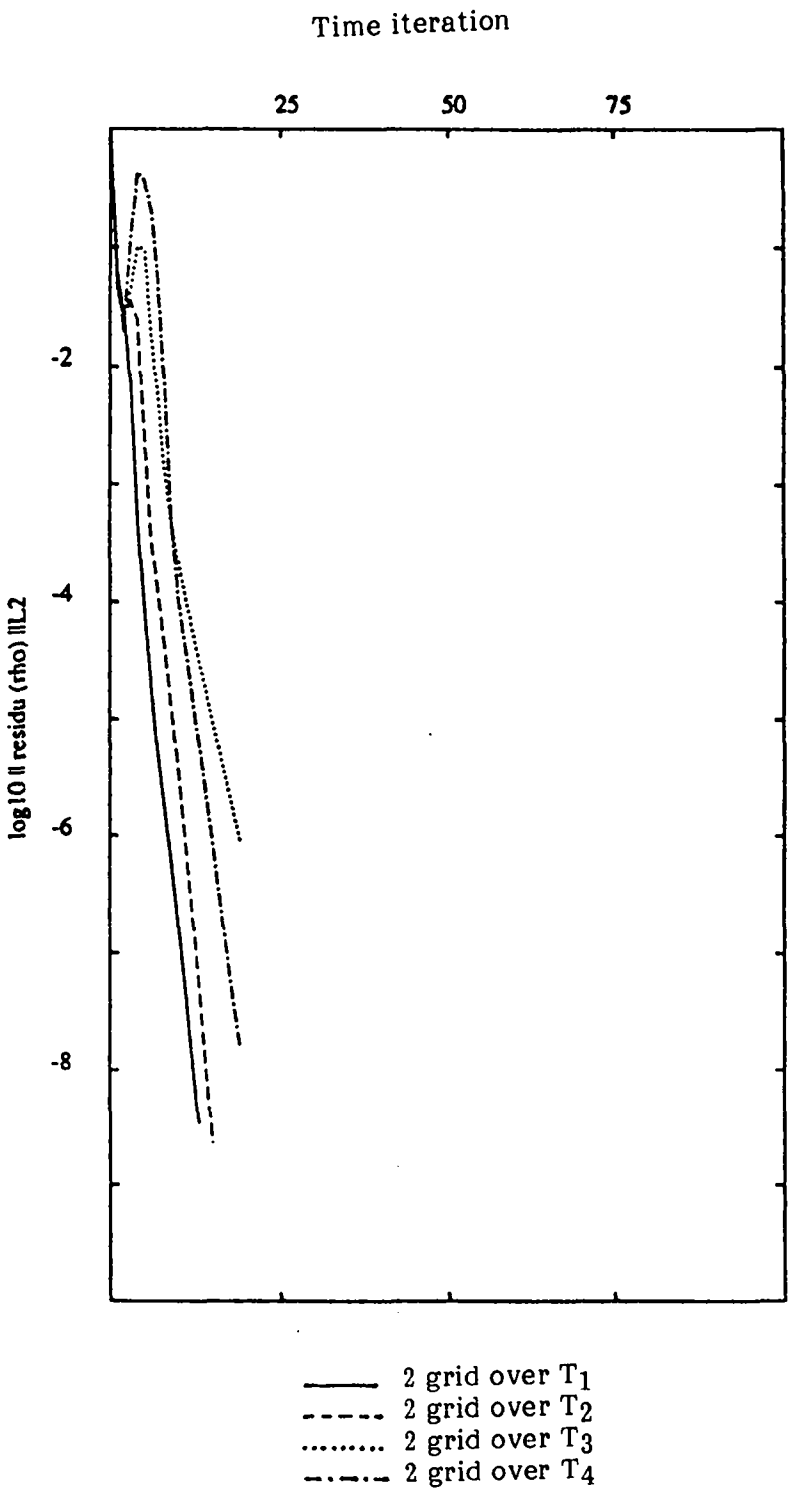


Figure 14c : Channel with bump ; Mach - .85.
First-order spatial approximation.
Nonlinear time convergence with CFL = 1000.

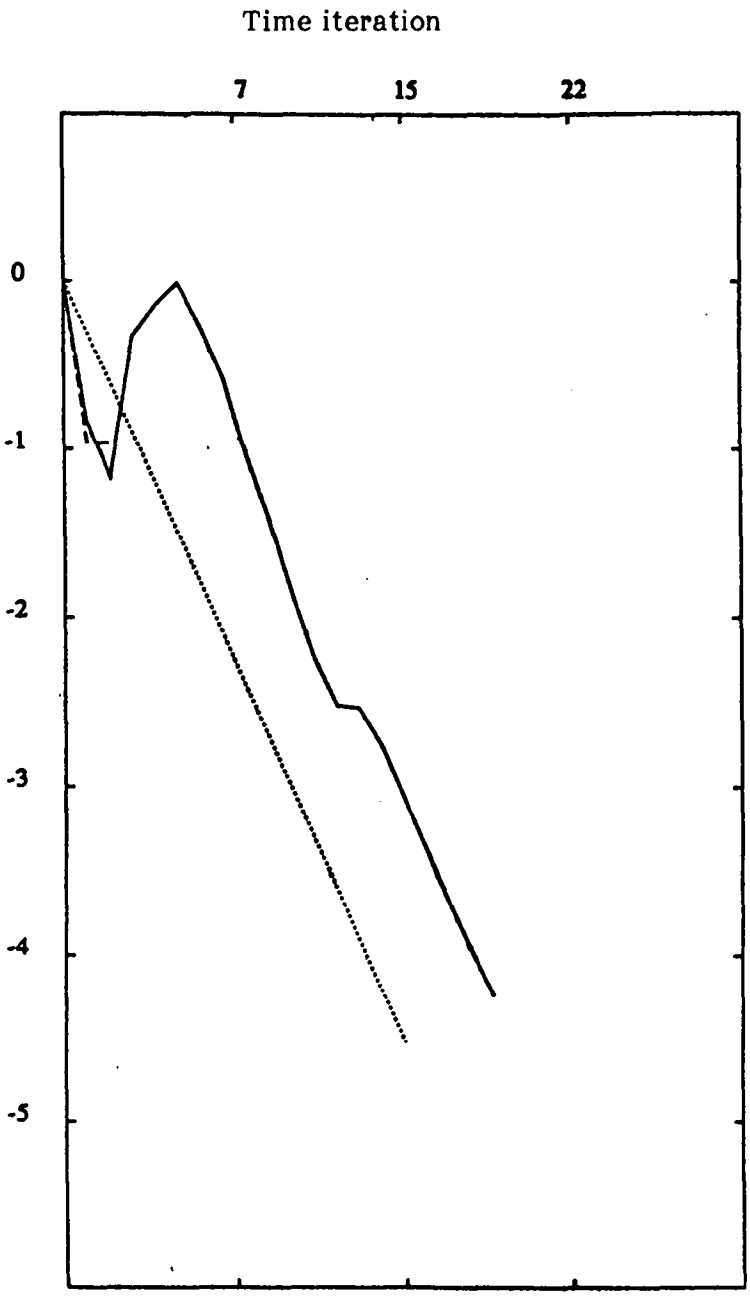


Figure 15 : Channel with bump. Mach = .85
second-order spatial approximation
Non linear time convergence, versus curve $n \rightarrow 2^{-n}$.

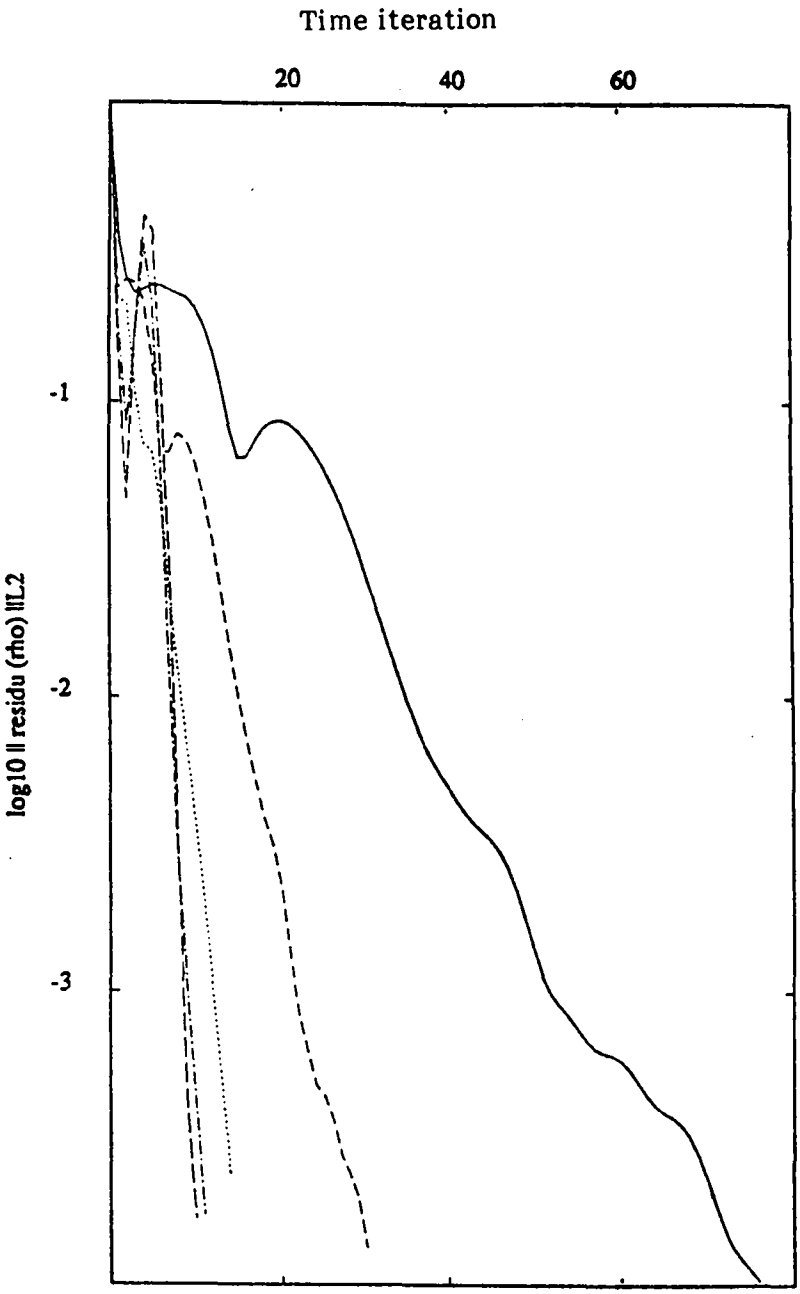


Figure 16 : Channel with bump ; Mach = .85
First-order spatial approximation
Linear phase with Jacobi- 4-grid iteration, $\omega = 1$
Non linear time convergence with CFL = 1000
(*comparison of various number of cycles in linear phase.*)

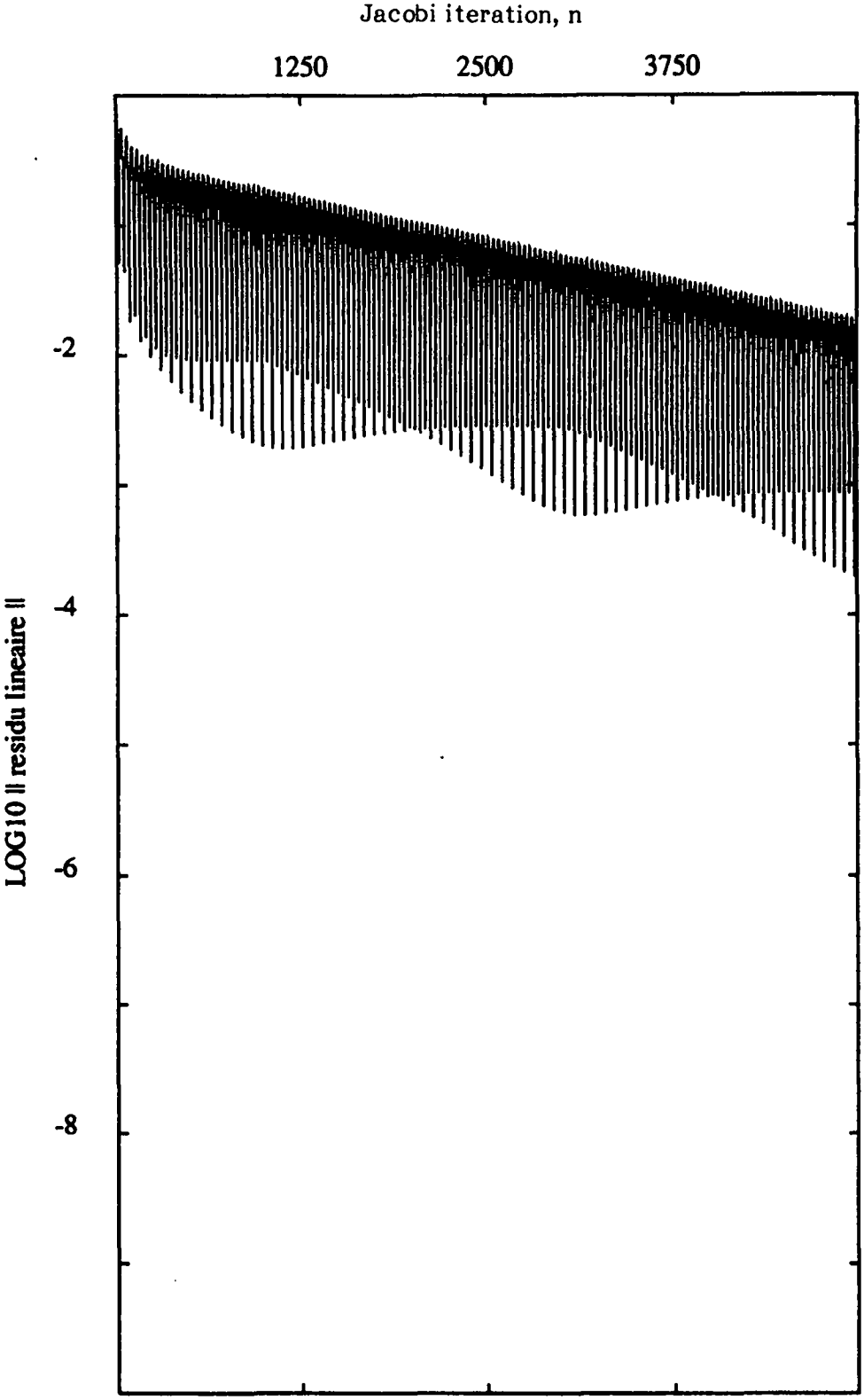


Figure 17 : Low Mach flow in a closed vessel
First time step (CFL = 100000) : convergence of the linear phase with Jacobi relaxation, $\omega = 1$

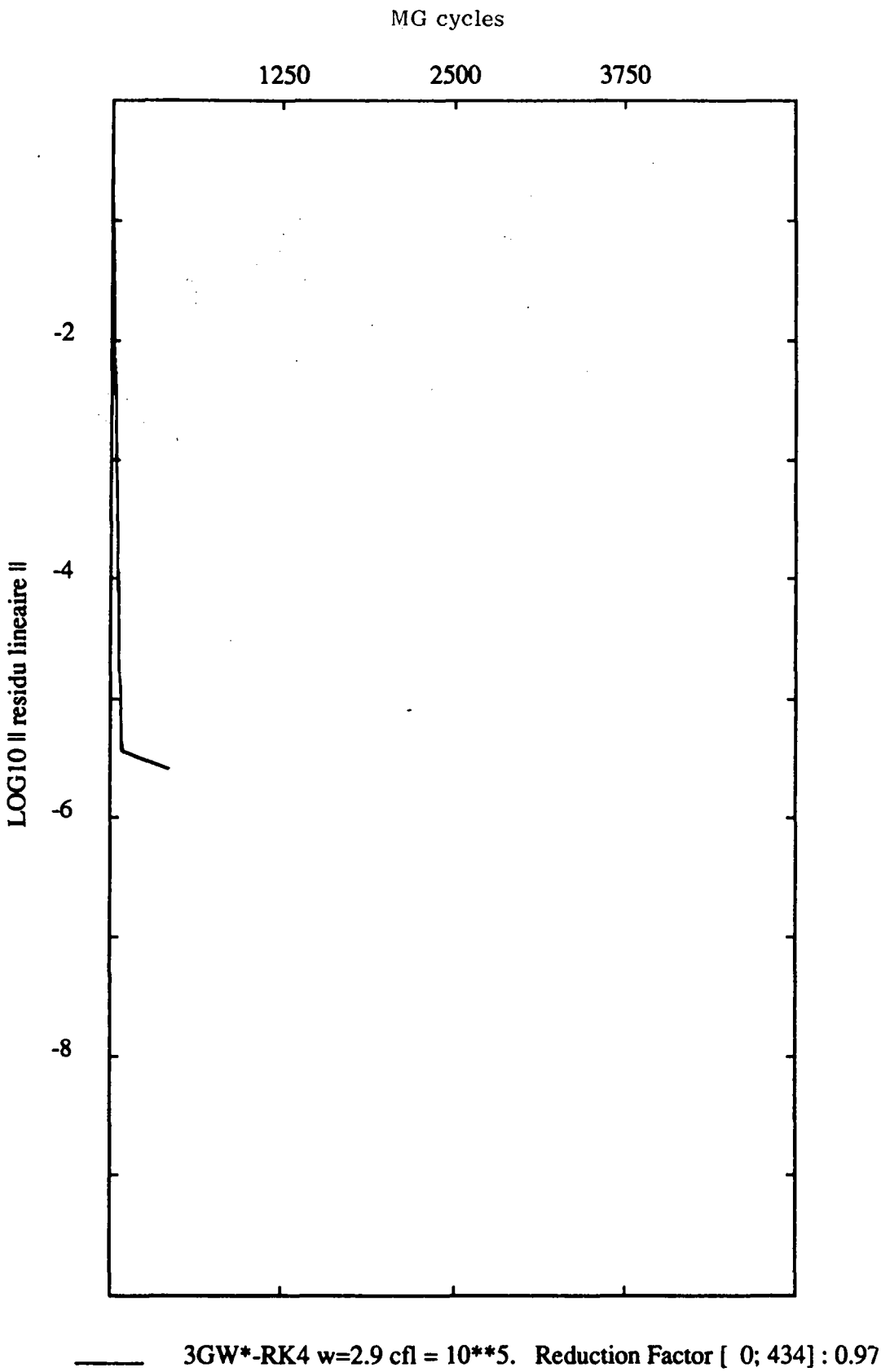


Figure 18 : Same as Figure 17, but linear phase solved with RK4–Jacobi MG cycles, $\omega = 2.9$, 434 cycles, mean reduction factor is .97

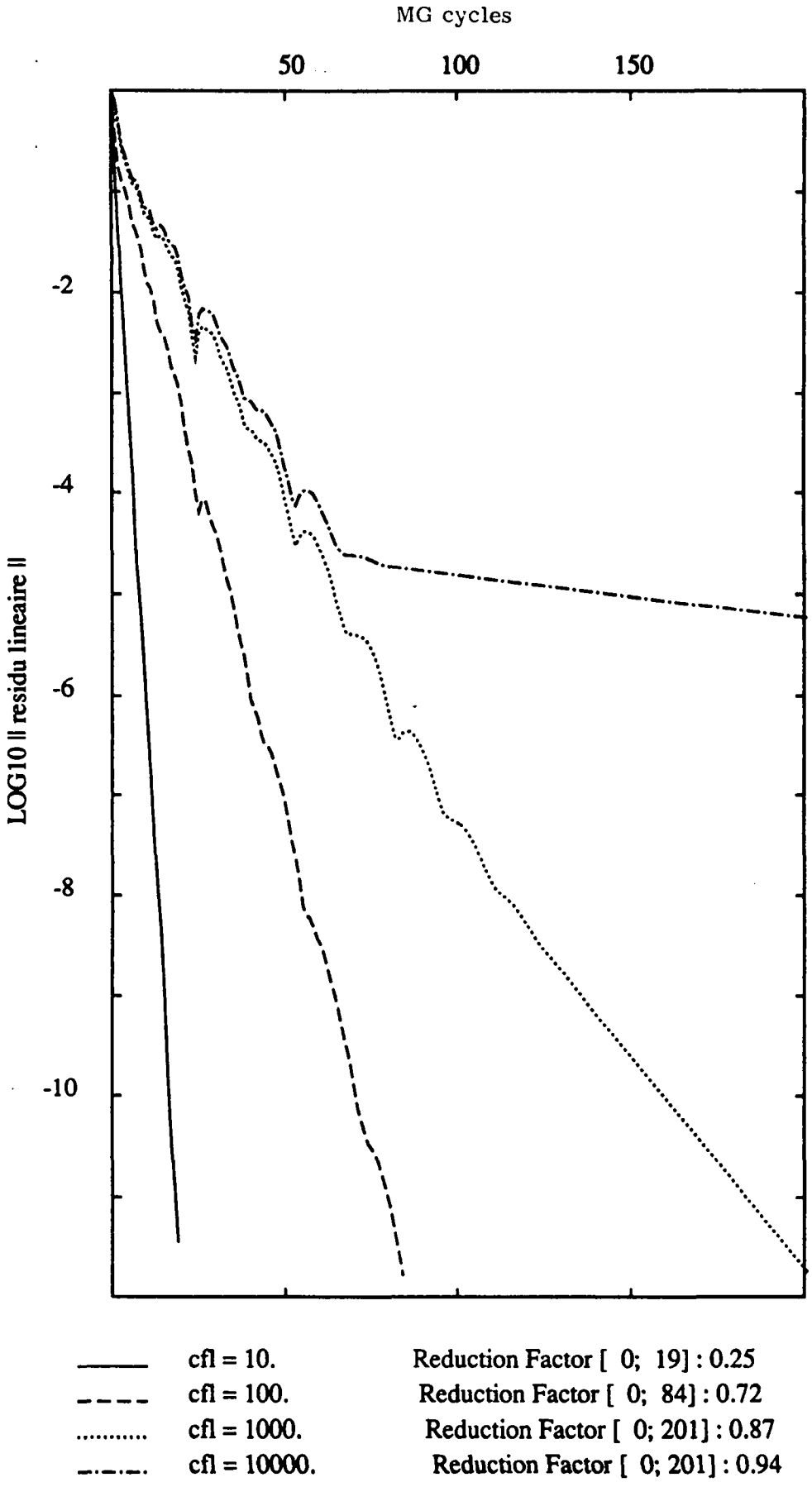


Figure 19 : Same as Figure 18, but CFL is varied.

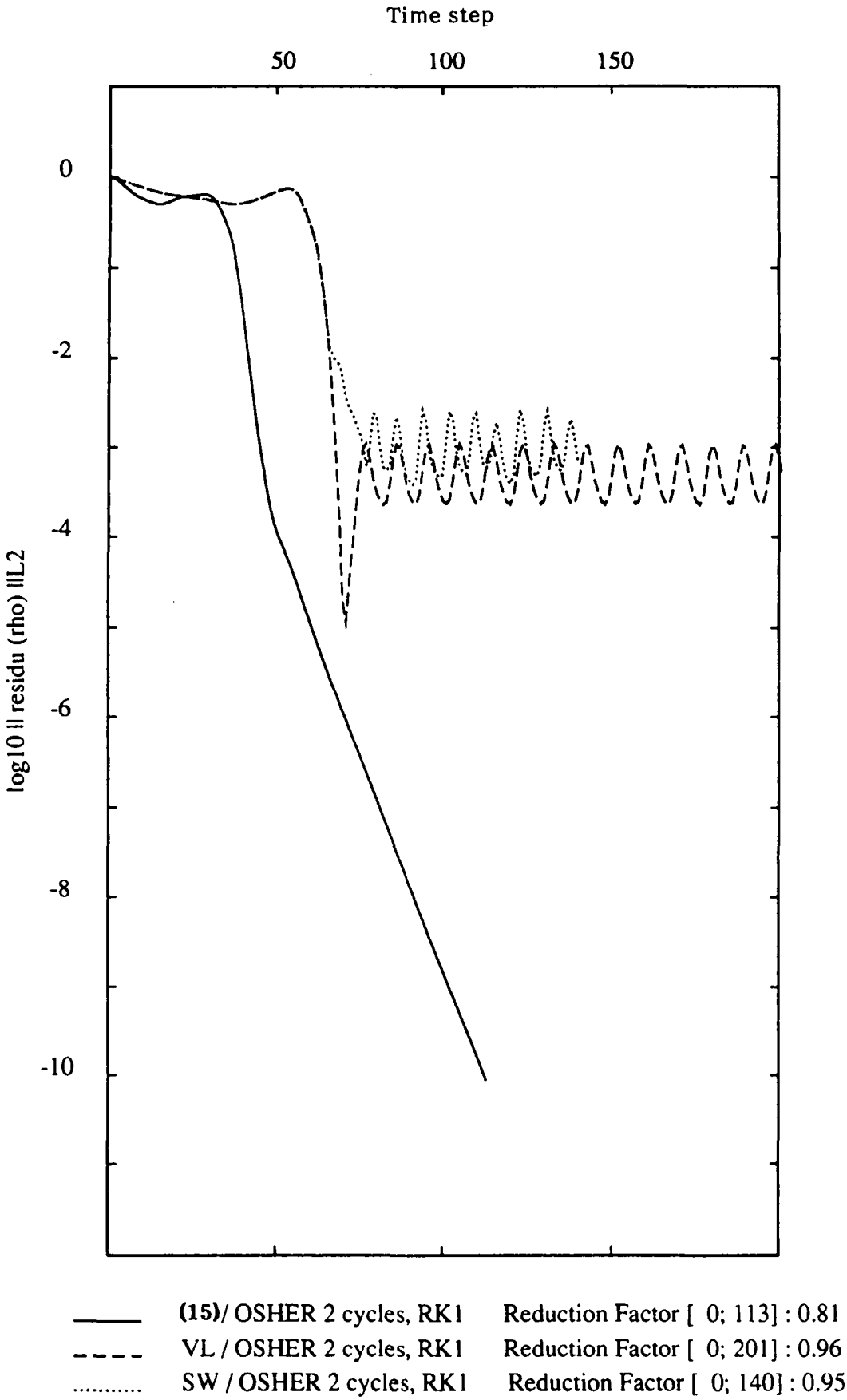


Figure 20 : Mach 8 flow around a NACA0012 airfoil ; 800 nodes : Comparison of three preconditioners, nonlinear time convergence from uniform flow (fastly increasing CFL number)

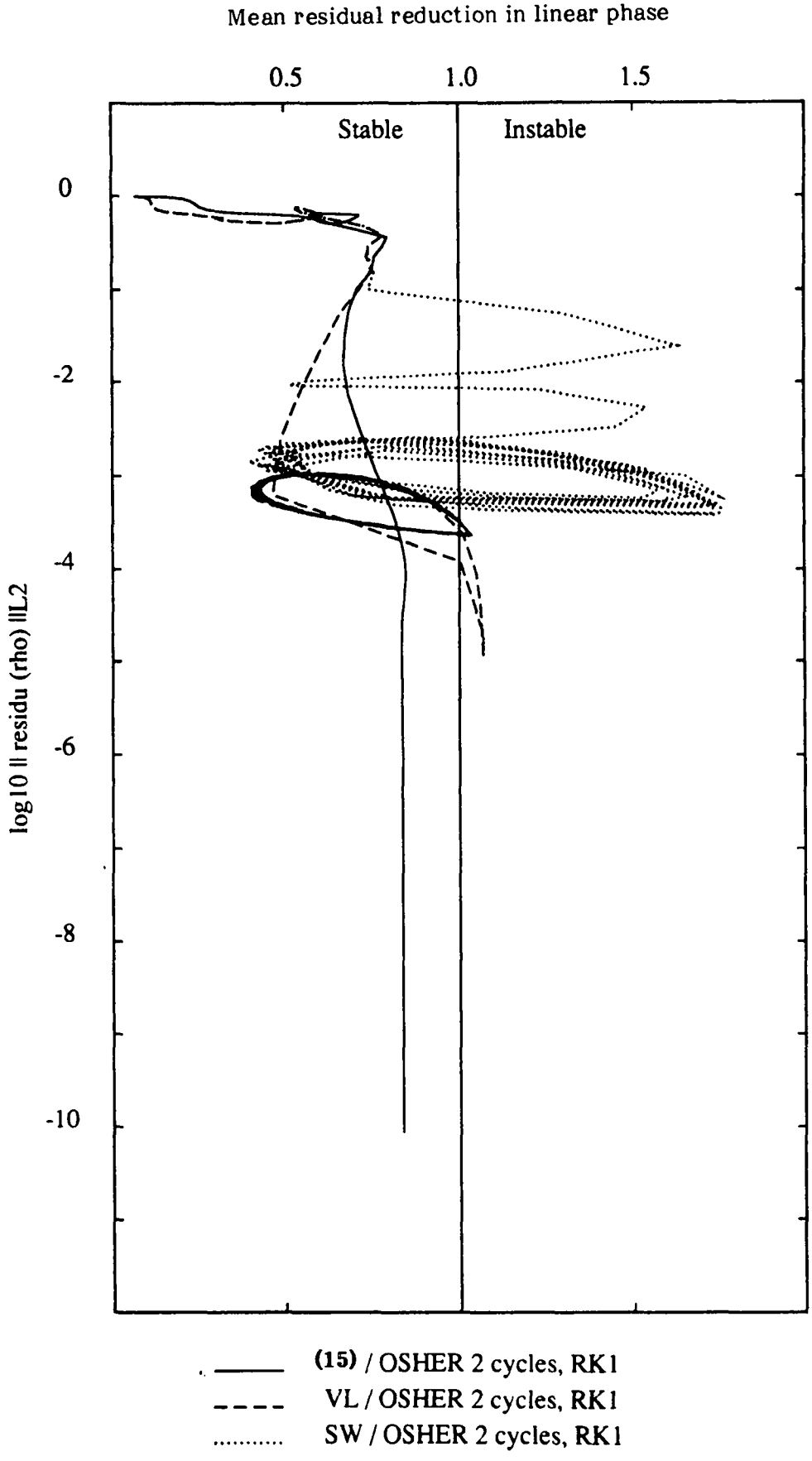


Figure 21 : Same as Figure 20 : convergence factor of the linear phase (along x) versus nonlinear residual (along y).

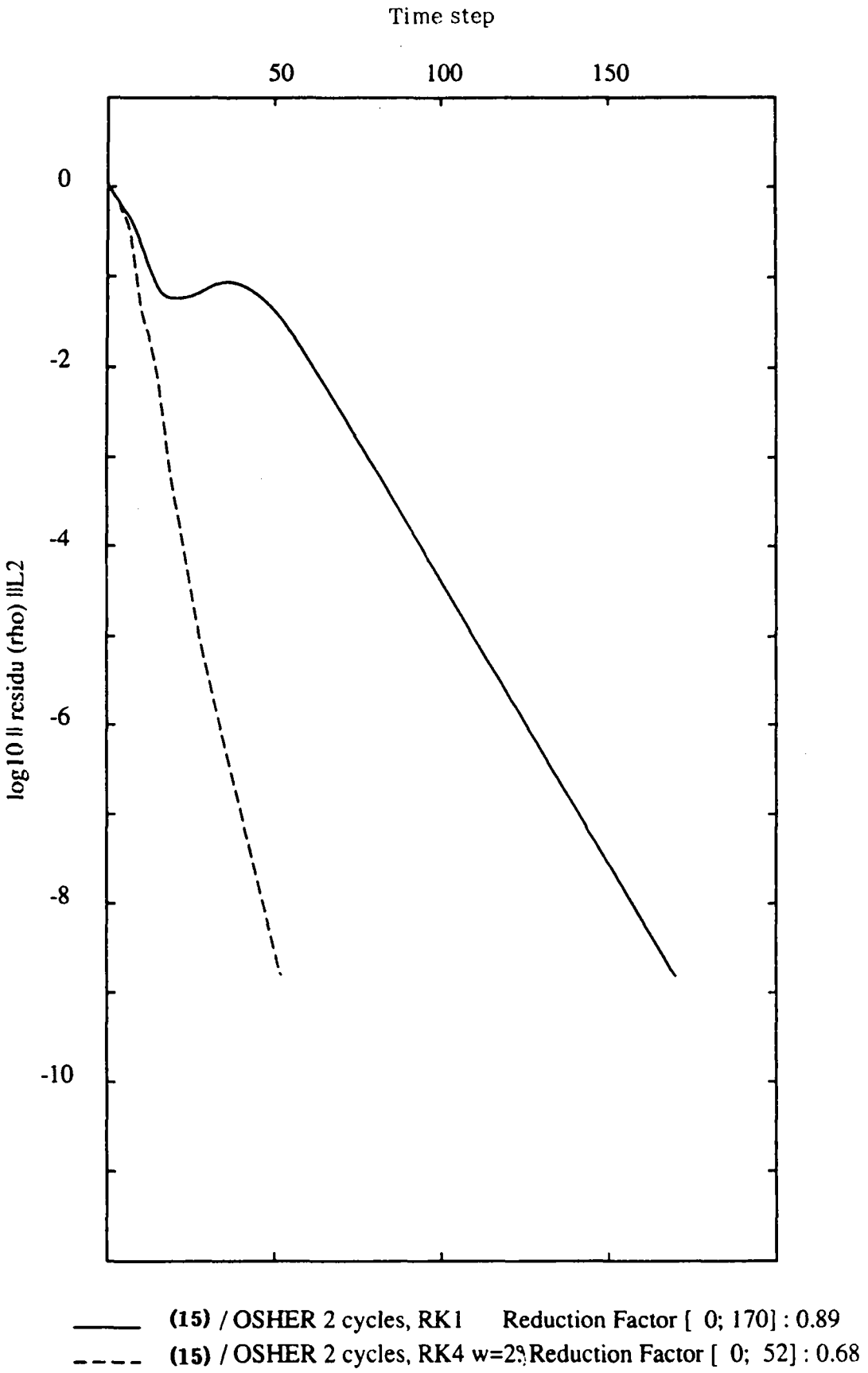


Figure 22 : Mach 8 flow around a NACA0012 airfoil, 3114 nodes, nonlinear time convergence from coarse–mesh solution (fastly increasing CFL). Comparison of one–grid/Jacobi versus 4-grid/RK4f-Jacobi, when two cycles are applied in the linear phase of each time iteration.

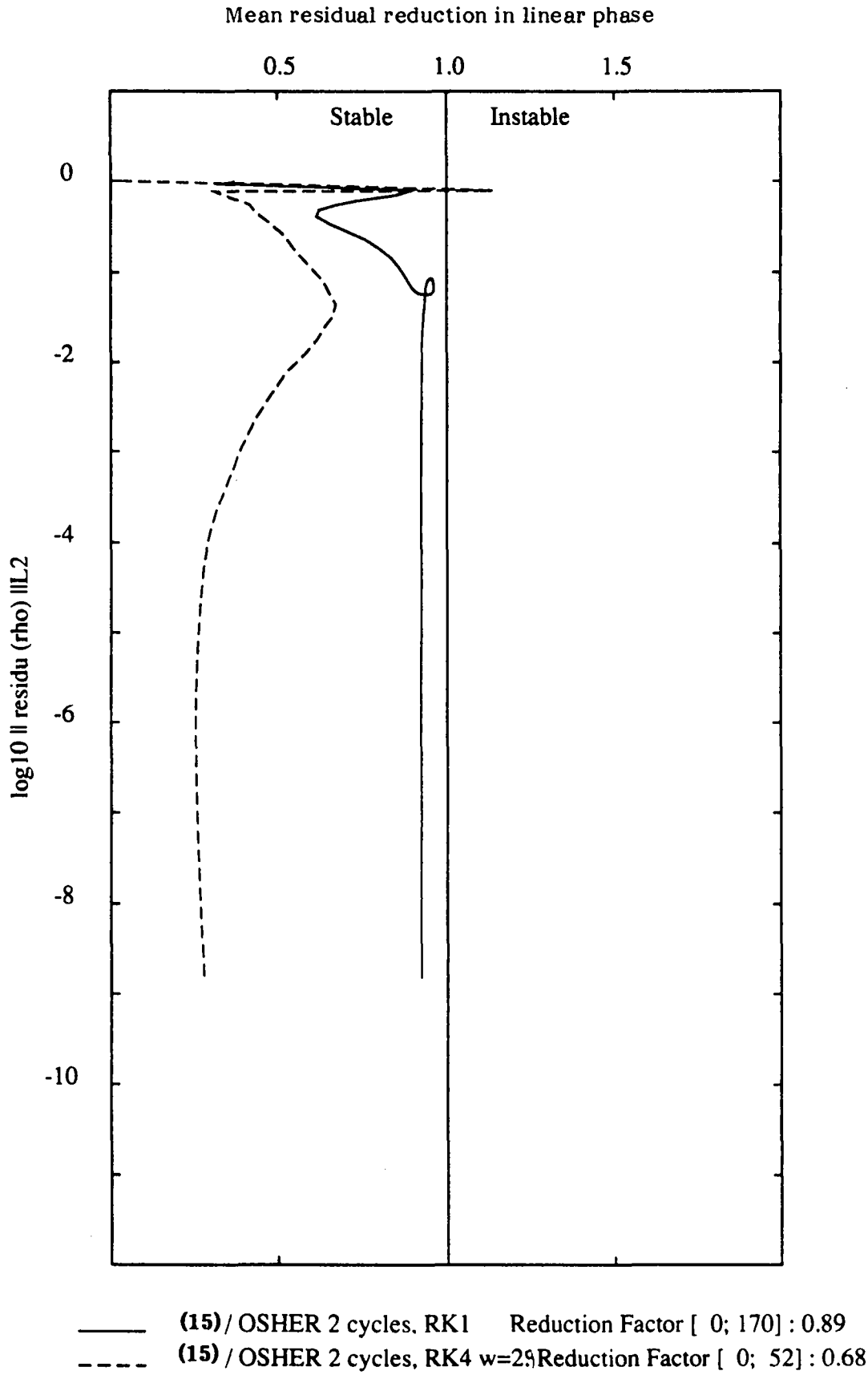


Figure 23 : Mach 8 flow around a NACA0012 airfoil, 3114 nodes, same computation as in Figure 22 ; convergence factor of the linear phase (along x) versus nonlinear residual (along y)

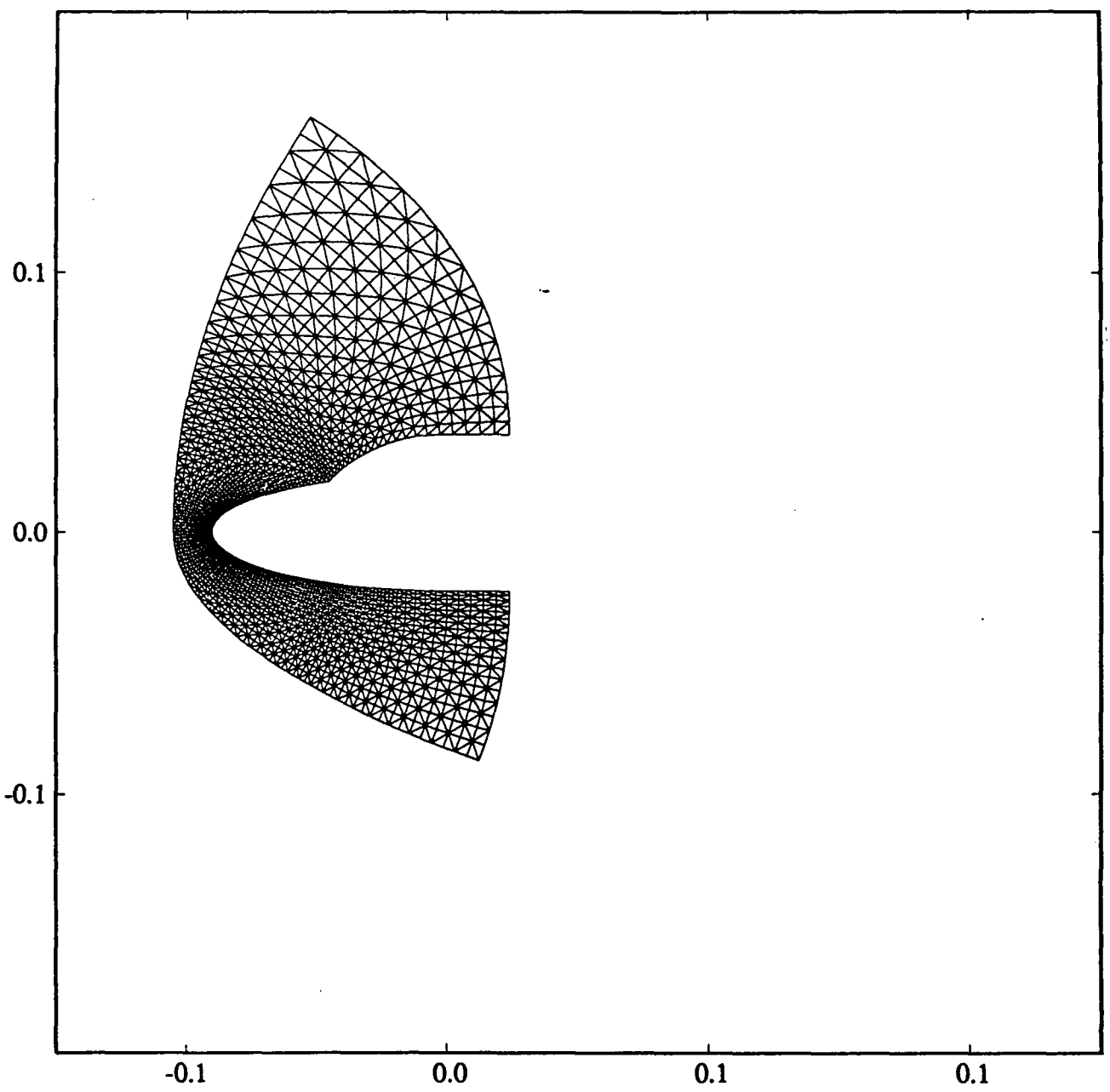


Figure 24 : Blunt body flow, Mach = 8 , 1700 nodes mesh

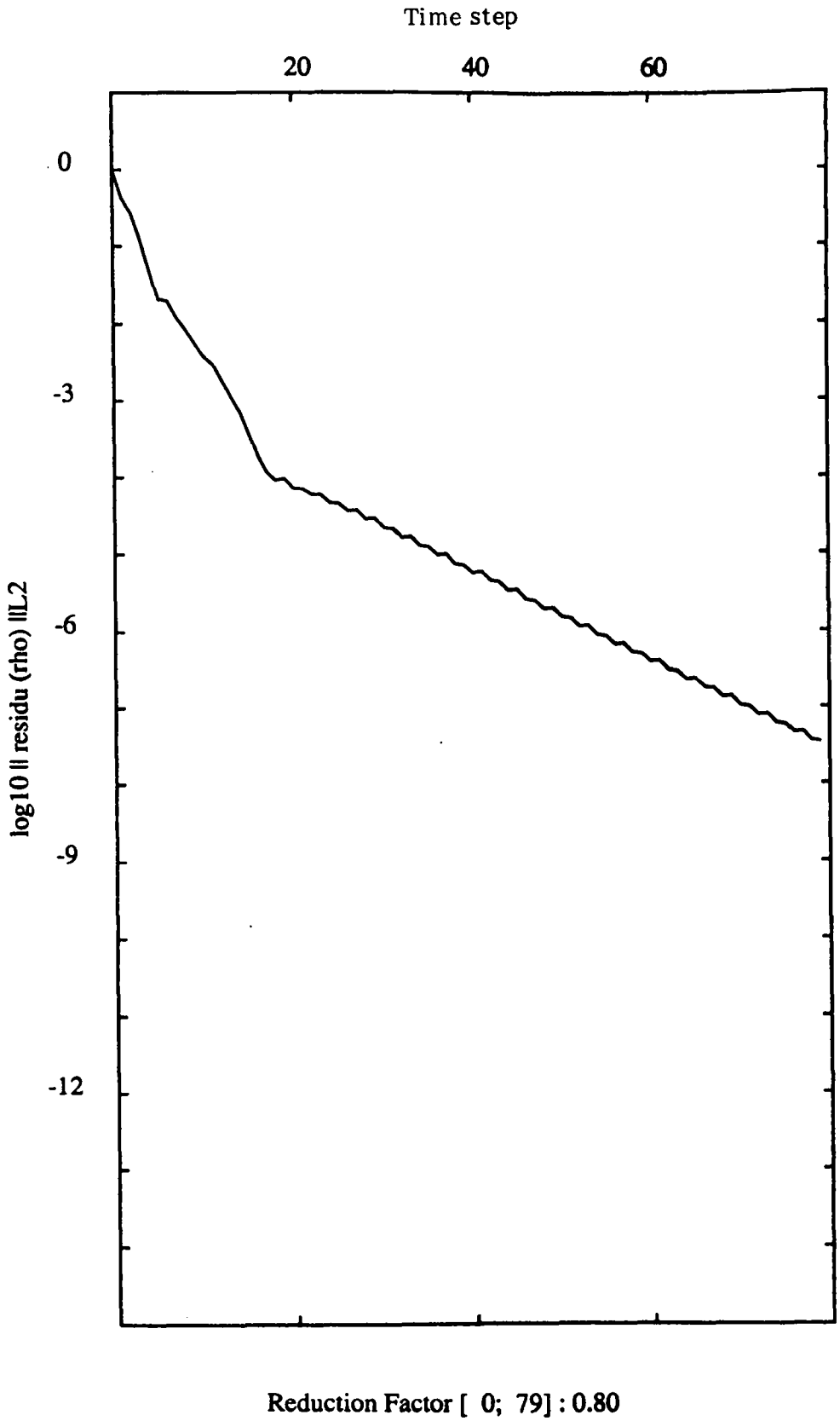


Figure 25 : Blunt body flow, Mach = 8, 1700 nodes, nonlinear convergence ; linear phase is solved with a 4-grid RK4 Jacobi ; initial condition was already converged by a factor of 10 000.

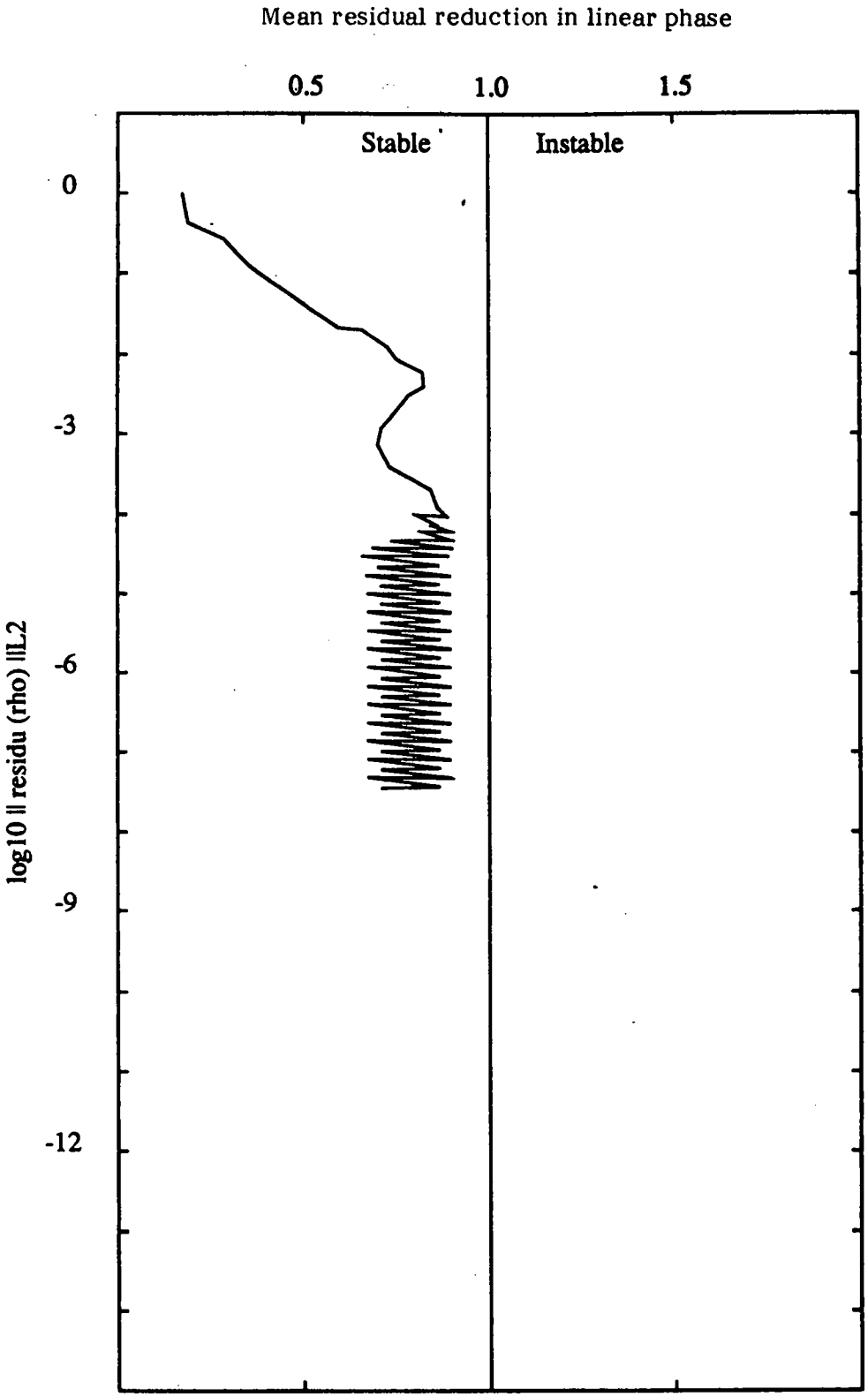


Figure 26 : Blunt body flow, Mach 8, 1700 nodes, same computation as in Figure 25 ; convergence factor of the linear phase (along x) versus nonlinear residual (along y)

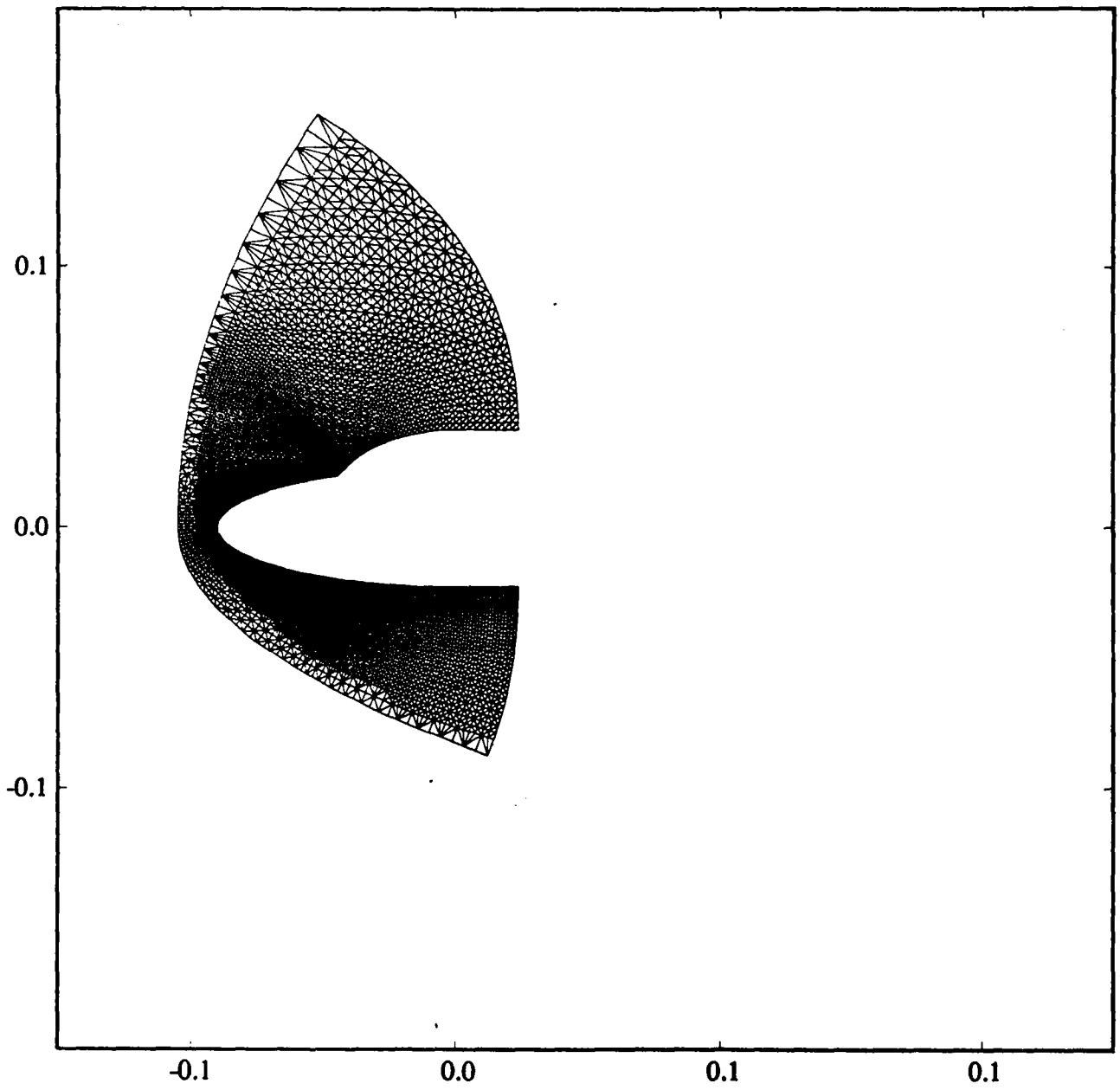


Figure 27 : Blunt body flow, Mach 8, 6000 nodes mesh

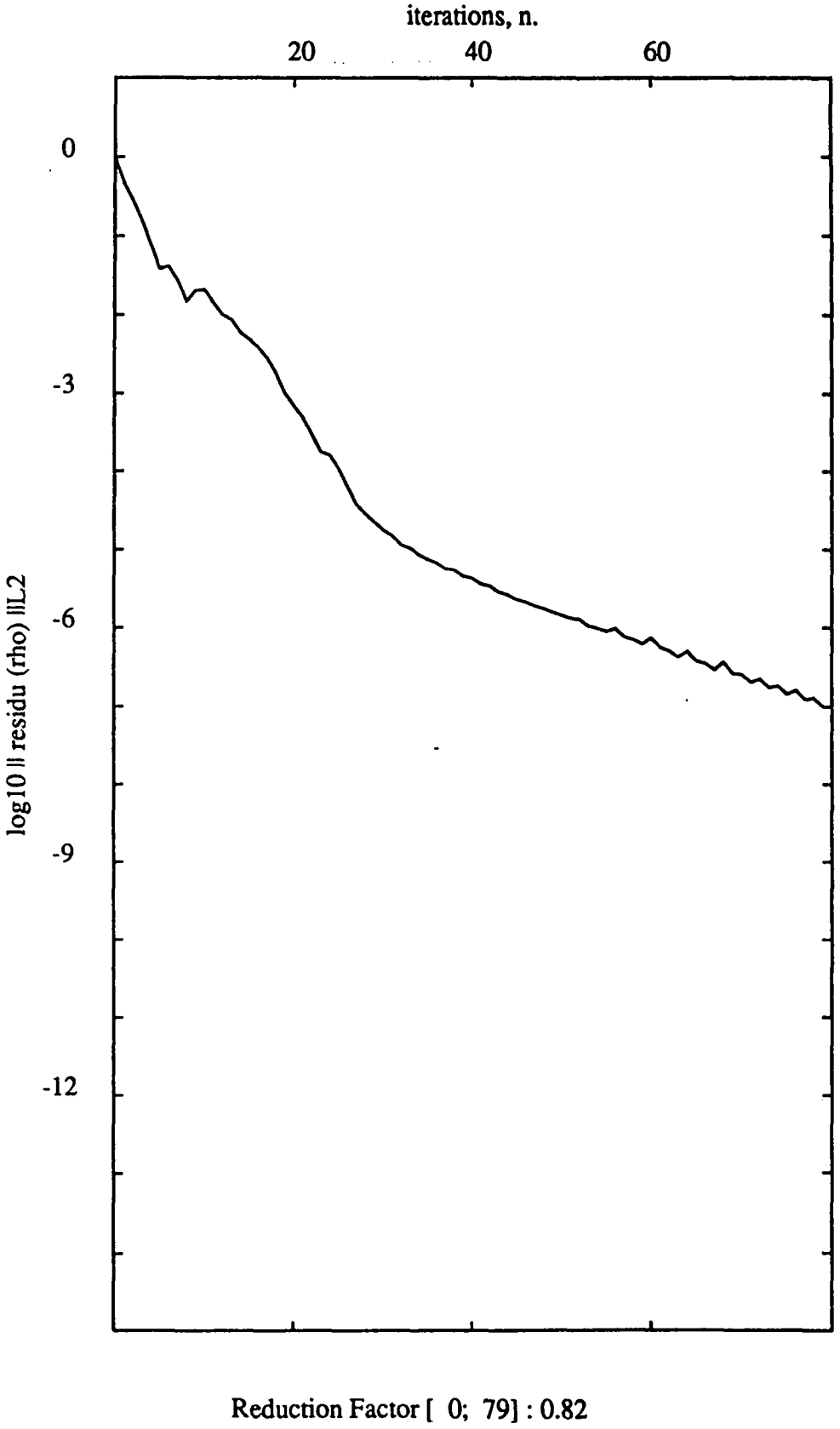


Figure 28 : Blunt body flow, Mach 8, 6000 nodes, nonlinear convergence ;
same parameters as in Figure 25 initial condition is interpolation of 1700-node solution

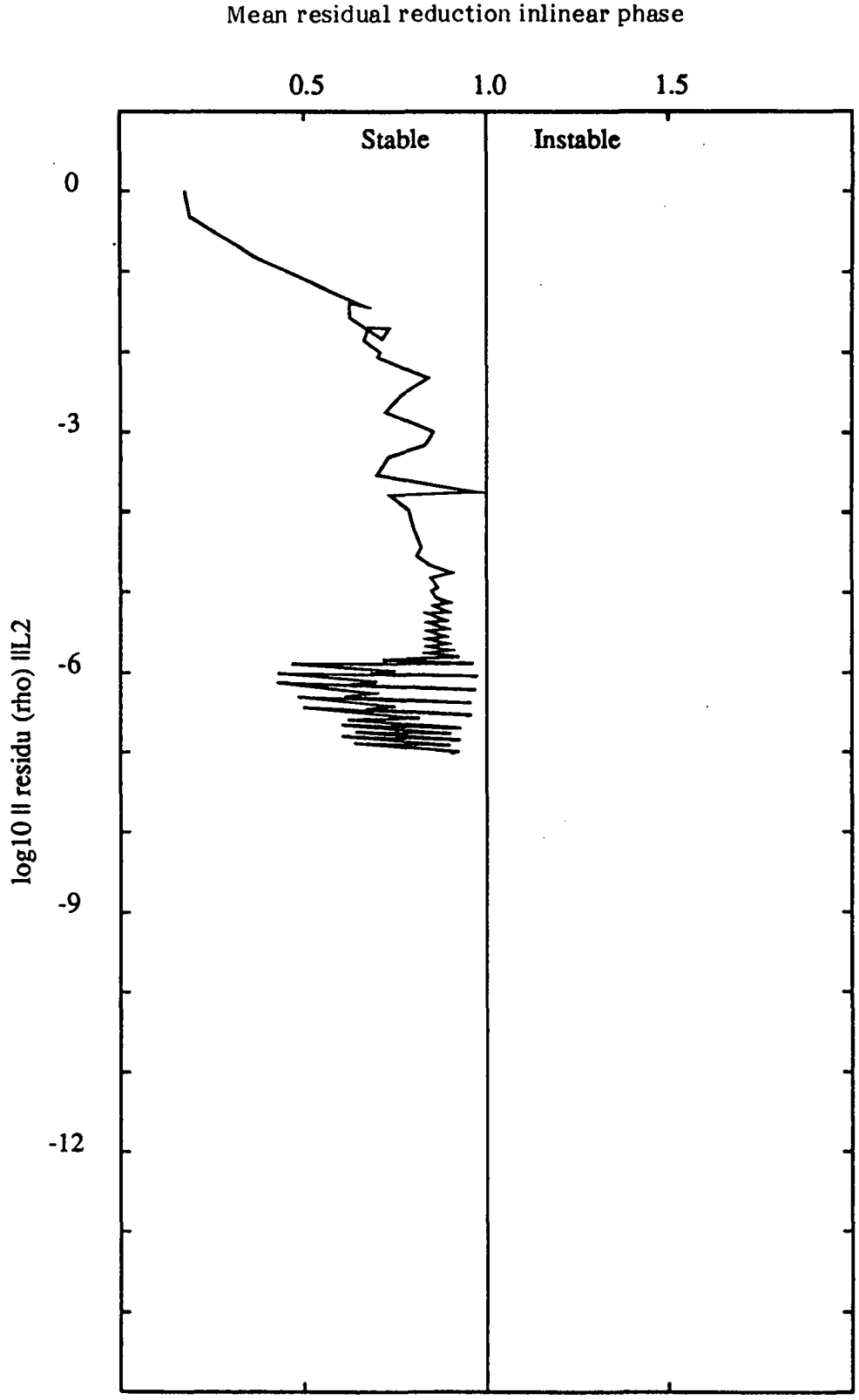


Figure 29 : computation as in Figure 28 ; convergence factor
of the linear phase (along x) versus nonlinear residual (along y)

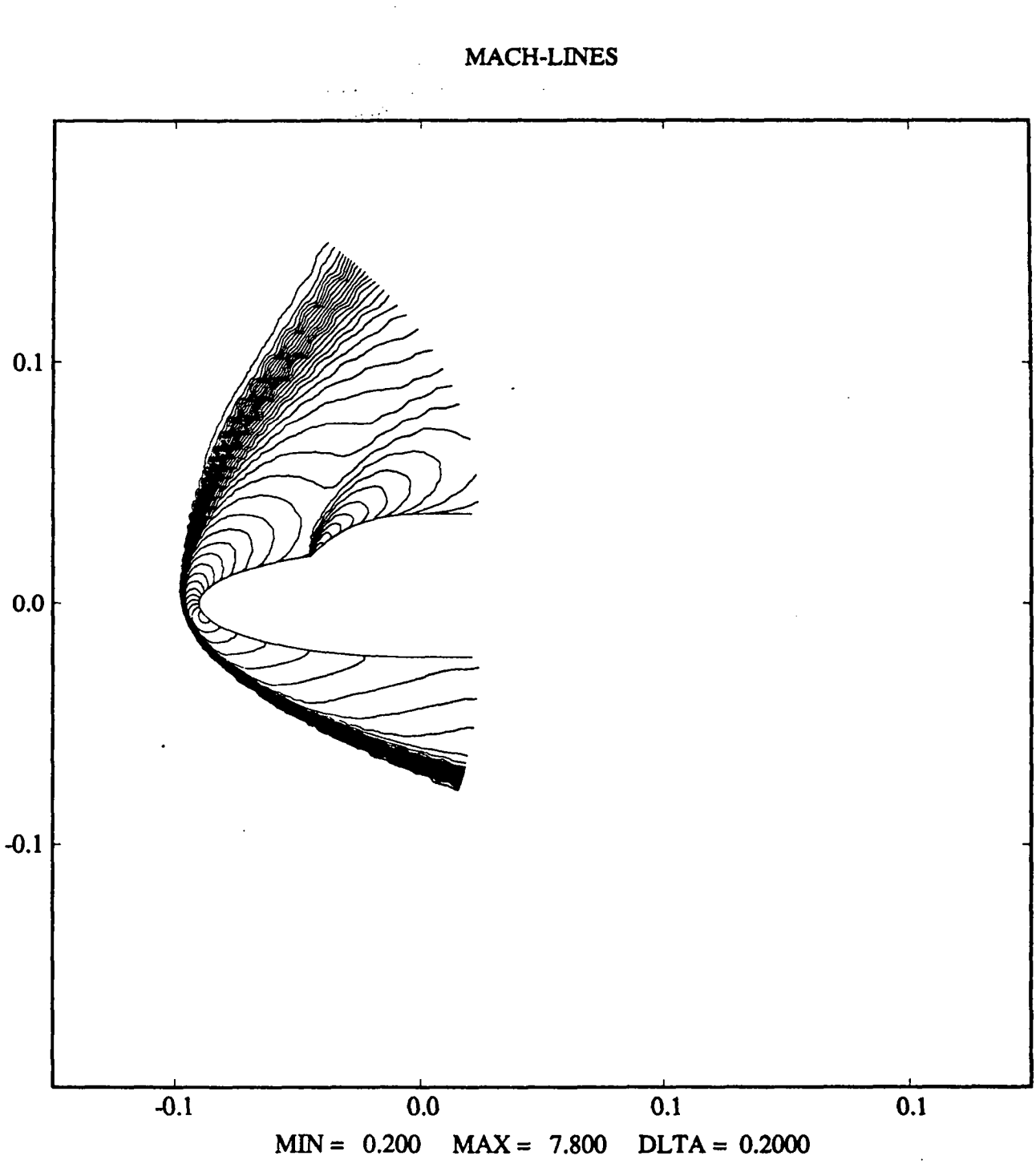


Figure 30 : Blunt body flow, Mach 8, 6000 nodes,
Mach contours from M = 0.2 to M = 7.8, increment = 0.2.

Imprimé en France
par
l'Institut National de Recherche en Informatique et en Automatique

ISSN 0249-6399